

UNCLASSIFIED

AD NUMBER

ADB007215

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; JUL 1975. Other requests shall be referred to Aeronautical Systems Division, Attn: YHCD, Wright-Patterson AFB, OH 45433.

AUTHORITY

ASD ltr 11 Jul 1977

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED  
AND CLEARED FOR PUBLIC RELEASE  
UNDER DOD DIRECTIVE 5200.20 AND  
NO RESTRICTIONS ARE IMPOSED UPON  
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED.



②

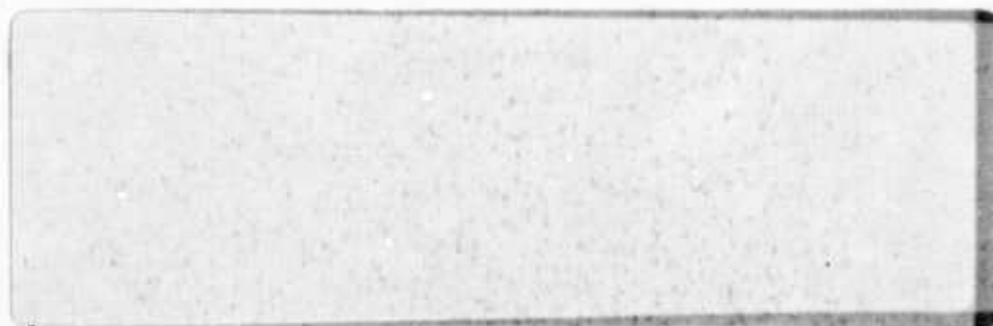
**Calspan**

# Technical Report

AD B 007215

AD No. \_\_\_\_\_  
DDC FILE COPY





SUBMISSION TO		White Section	<input type="checkbox"/>
BY		Self Section	<input type="checkbox"/>
NTS			
SEC			
IDENTIFICATION			
DISTRIBUTION/AVAILABILITY			
Dist.	Final, Int. or Special		
<b>B</b>			

✓

2

# Calspan

B-1 SYSTEMS APPROACH TO TRAINING  
TECHNICAL MEMORANDUM SAT-6 ✓

TRAINING RESOURCES ANALYTICAL MODEL (TRAM)  
PROGRAMMER'S MANUAL

JULY 1975

Distribution limited to U.S. Government Agencies only; test and evaluation; July 1975. Other requests for this document must be referred to B-1 System Program Office, ASD/YHCD, Wright-Patterson Air Force Base, Ohio, 45433.



PREPARED BY:

William F. H. Ring  
William F. H. Ring

APPROVED BY:

Robert C. Sugarman  
Robert C. Sugarman  
SAT Task Manager

George Gaidasz  
George Gaidasz

APPROVED BY:

D. Barry Dahm  
D. Barry Dahm, Head  
Environmental & Energy  
Systems Department

John R. Menig  
John R. Menig

Walter L. Stortz  
Walter L. Stortz

APPROVED BY:

Robert C. Kidder for  
Robert C. Kidder  
Program Manager  
B-1 Technical Support Program

CALSPAN CORPORATION  
CONTRACT NO. F33657-75-C-0021

Calspan Corporation  
Buffalo, New York 14221

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) B-1 Systems Approach to Training. <del>Blank Report</del> Training Resources Analytic Model (TRAM). Programmer's Manual.		5. TYPE OF REPORT & PERIOD COVERED Final Report Technical Memorandum July 1974 - October 1975 6. PERFORMING ORG. REPORT NUMBER SAT-6 ✓
7. AUTHOR(s) William F.H. Ring, Walter L. Stortz George Gaidasz John R. Menig		8. CONTRACT OR GRANT NUMBER(s) F33657-75-C-0021 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS Calspan Corporation P. O. Box 235 Buffalo, New York 14221 ✓		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Aeronautical Systems Division B-1 Systems Project Office Wright-Patterson Air Force Base, OH 45433		12. REPORT DATE 11 JULY 1975 13. NUMBER OF PAGES 640
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 14 CALSPAN-TM-SAT-6		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to U.S. Government Agencies only: test and evaluation; July 1975. Other requests for this document must be referred to B-1 Systems Program Office, ASD/YHCD, Wright-Patterson Air Force Base, Ohio 45433.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Instructional Systems Development      Resource Management Systems Approach to Training      Computer Storage B-1      Training Economic Modeling      Training Resources		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The TRAM is a multiphase set of computer programs. Each program ex- cept for the sorting steps is described by a user's guide and programmer's guide. This document is the Programmer's Guide. The programmer's guide is intended to be a supplement to the user's guide, thereby saving unnecessary repetition except where the repetition is useful for understanding the material.		

(AD-3007 214) → (Over)

407 727

Unclassified

20. (Continued)  
or P 1473B

→ The purpose of Phase 1 is to assemble most of the data and to check it for consistency and completeness. The result of Phase 1 is normally a tape which is passed to Phase 2. Phase 2 makes further checks on linkages and network integrity. Phase 2 prepares lists of names, student demands, trainee source lists and resource lists. Phase 3 resolves the trainee demands into classes and determines the amount of resources used by simulating the training system. The output of Phase 3 consists of source and lag records which indicate the occurrence of trainee matriculation, lags due to lack of resources, and an unused resources file. Phase 4 computes the amount of resources used by comparing the unused and original resources, and then prepares an economic analysis of the run. Phase 5 processes the trainee source and lag records and writes a report on these uses.

TROLIE is a quick-look version of Phases 1,2 and 3 of TRAM which can be used for less detailed analysis.

X

Unclassified

1473B



## PREFACE

This document is one of several technical memoranda which have been delivered to the B-1 Systems Project Office (B-1 SPO) in performance of the Systems Approach to Training (SAT) Task under Contract Number F33657-75-C-0021. Each of the separate SAT documents is listed below. Additional copies may be requested from: B-1 Systems Project Office, Data Configuration Division, Wright-Patterson Air Force Base, Ohio.

<u>Technical Memoranda</u>	<u>Number</u>	<u>Author(s)</u>	<u>Date</u>
B-1 Systems Approach to Training, Final Report.	SAT- 1 Vol. 1	R. Sugarman S. Johnson W. Ring	July 1975
B-1 Systems Approach to Training, Final Report. Appendix A: Cost Details.	SAT- 1 Vol. 2	H. Reif W. Ring	July 1975
B-1 Systems Approach to Training, Final Report. Appendix B: Bibliography and Data Collection Trips.	SAT- 1 Vol. 3	A. Blair	July 1975
Behavioral Objectives for the Pilot, Copilot, and Offensive Systems Operator.	SAT- 2 Vol. 1 & 2	J. Mitchell W. Hinton S. Johnson	July 1975
Simulation Technology Assessment Report (STAR).	SAT- 3	S. Johnson J. Knight R. Sugarman	July 1975
Sorting Model for B-1 Aircrew Training Data. User's and Programmer's Guide.	SAT- 4	J. Menig T. Ranney	July 1975
Training Resources Analytic Model (TRAM). User's Manual.	SAT- 5	W. Ring G. Gaidasz J. Menig W. Stortz	July 1975
Training Resources Analytic Model (TRAM). Programmer's Manual.	SAT- 6	W. Ring G. Gaidasz J. Menig W. Stortz	July 1975
Task Analysis Listings.	SAT- 7	J. Mitchell T. Ranney	July 1975
Control/Display Catalog and Action Verb Thesaurus.	SAT- 8	T. Ranney A. Blair	July 1975

July 1975  
SAT-6

TRAINING RESOURCES ANALYTIC MODEL  
(TRAM)  
PROGRAMMER'S MANUAL

William F.H. Ring  
George Gaidasz  
John R. Menig  
Walter L. Stortz

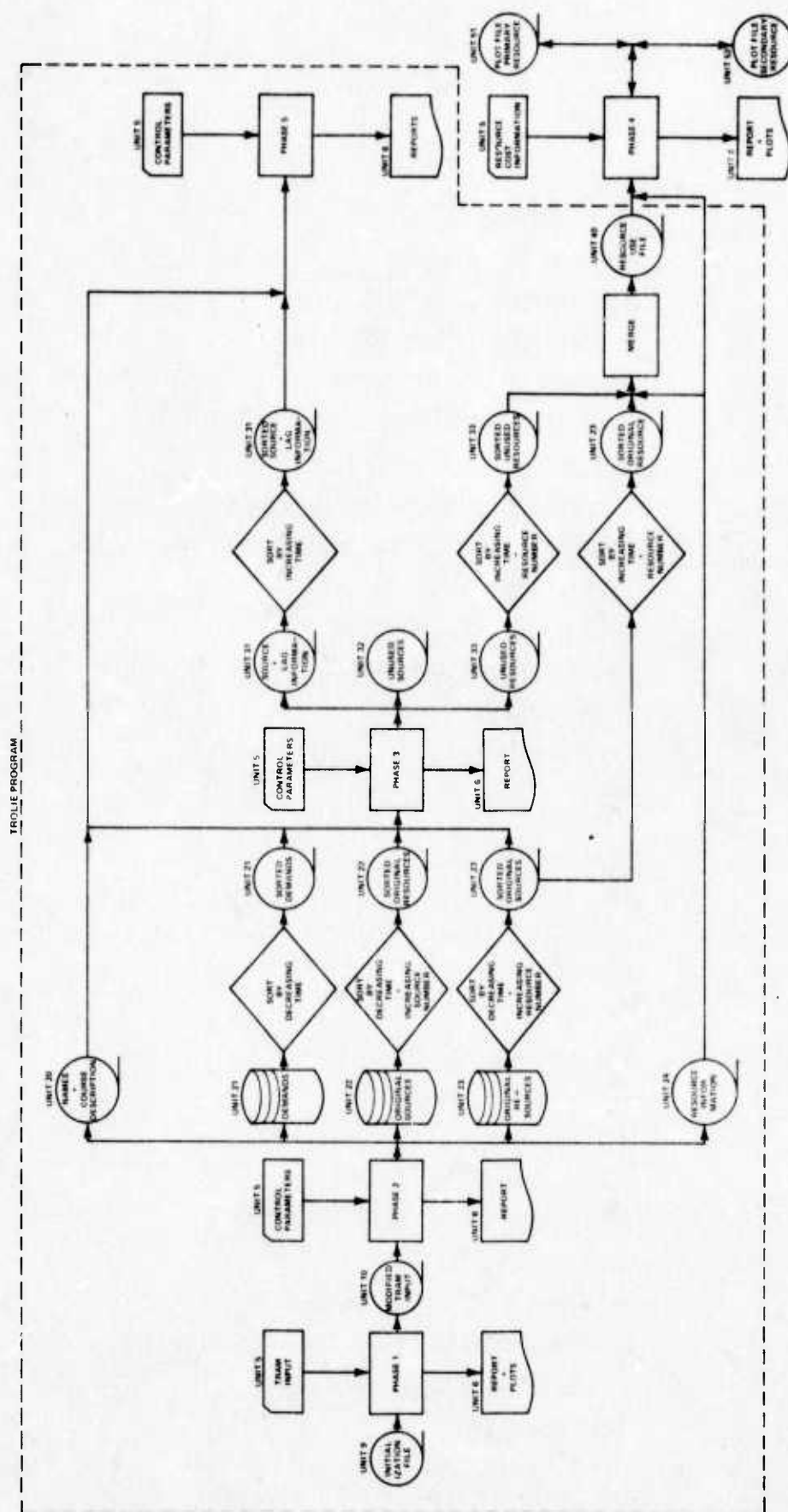
SUMMARY

The TRAM is a multiphase set of computer programs. The following figure is the overall flow diagram which depicts each program within TRAM and the data sets associated with it. Each program except for the sorting steps is described by a user's guide and programmer's guide. This document is the Programmer's Guide.

The programmer's guide is intended to be a supplement to the user's guide thereby saving unnecessary repetition except where the repetition is useful for understanding the material.

The purpose of Phase 1 is to assemble most of the data and to check it for consistency and completeness. The result of Phase 1 is normally a tape which is passed to Phase 2. Phase 2 makes further checks on linkages and network integrity. Phase 2 prepares lists of names, student demands, trainee source lists and resource lists. Phase 3 resolves the trainee demands into classes and determines the amount of resources used by simulating the training system. The output of Phase 3 consists of source and lag records which indicate the occurrence of trainee matriculation, lags due to lack of resources, and an unused resources file. Phase 4 computes the amount of resources used by comparing the unused and original resources, and then prepares an economic analysis of the run. Phase 5 processes the trainee source and lag records and writes a report on these uses.

TROLIE is a quick-look version of Phases 1, 2 and 3 of TRAM which can be used for less detailed analysis.



## TRAM OVERALL ORGANIZATION

(Dashed lines indicate portions of TRAM replaced by TROLE-the quick-look program)



# TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
1.1	INTRODUCTION	1
1.2	PROGRAM DESCRIPTION	2
1.3	DESCRIPTION OF INPUTS	4
1.4	DESCRIPTION OF OUTPUTS	6
1.5	SUBPROGRAM DESCRIPTIONS	7
1.6	SUBROUTINE CROSS REFERENCE TABLE	74
1.7	COMMON VARIABLE DEFINITION	83
1.8	INTERNAL DATA BLOCK DESCRIPTION	89
1.9	COMMON VARIABLE CROSS REFERENCE TABLE	105
1.10	INITIALIZATION FILE	112
1.11	OUTPUT FILE DESCRIPTION	113
1.12	PHASE 1 ERROR MESSAGES	130
2.0	TRAM PHASE 2	133
2.1	FLOW CHARTS	134
2.2	DESCRIPTIONS OF RECORDS AND VARIABLES USED IN COMMONS	167
2.3	DESCRIPTION OF ROUTINES	188
2.4	CROSS REFERENCE TABLES FOR ROUTINES AND VARIABLES USED IN COMMONS	234
3.0	TRAM PHASE 3	260
3.1	INTRODUCTION	260
	3.1.1 DATA MANAGEMENT	260
3.2	DESCRIPTION OF INPUTS	260
3.3	DESCRIPTION OF OUTPUTS	266
3.4	SUBPROGRAM DESCRIPTIONS	271
3.5	SUBPROGRAM CROSS REFERENCE TABLE	423
3.6	COMMON VARIABLE DEFINITIONS	444
3.7	INTERNAL DATA BLOCK DESCRIPTIONS	469
3.8	COMMON VARIABLE CROSS REFERENCE TABLE	476
4.0	PHASE 4 PROGRAMMER'S GUIDE	535
4.1	INTRODUCTION	535
4.2	PROGRAM DESCRIPTION	536

TABLE OF CONTENTS (CONT)

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
4.3	SUBPROGRAM DESCRIPTIONS	538
4.4	SUBROUTINE CROSS REFERENCE TABLE	581
4.5	COMMON VARIABLE DEFINITIONS	589
4.6	COMMON VARIABLE CROSS REFERENCE TABLE	598
4.7	TEMPORARY FILES	602
5.0	PHASE 5 PROGRAMMER'S GUIDE	605
5.1	INTRODUCTION	605
5.2	PROGRAM DESCRIPTION	606
5.3	SUBPROGRAM DESCRIPTIONS	607
5.4	SUBROUTINE CROSS REFERENCE TABLE	613
5.5	COMMON VARIABLE DEFINITIONS	615
5.6	COMMON VARIABLE CROSS REFERENCE TABLE	618
6.0	MERGE PROGRAMMER'S GUIDE	621
6.1	INTRODUCTION	621
6.2	PROGRAM DESCRIPTION	622
6.3	DESCRIPTION OF INPUTS	626
6.4	DESCRIPTION OF OUTPUTS	627
7.0	TROLIE PROGRAMMER'S GUIDE	628
7.1	INTRODUCTION	628
7.2	SUBROUTINES	628
7.3	NOTES	629
7.4	COMMONS	630
7.5	REPORTS	632
7.6	DATA SET OUTPUTS	632

## Section 1.1

### INTRODUCTION

The purpose of phase 1 is to read the user inputs and convert them to the internal format required for phase 2. It also tests the inputs for validity and provides the necessary outputs to document the run.

The manual is intended to aid the programmer in the operation and modification of the computer program by supplementing the users' guide.

## Section 1.2

### PROGRAM DESCRIPTION

Phase 1 performs the following functions:

- reads the input cards and prints them
- checks all values for validity
- prints formatted tables of the inputs
- replaces user assigned names with an internal ID number
- plots a course block diagram
- sorts the data records and writes them onto unit 10 for phase 2.

Phase 1 initializes certain common areas with data contained on its initialization file (unit 9). The values on this file are considered program constants, rather than inputs, and any changes to them are usually accompanied with program modifications. See Section 1.0 for a description of this file.

The values from each input card are printed as they are read. Also, each numeric value is tested against a range of acceptable values to see if it is valid. If it is not, a diagnostic message is printed, which will appear immediately after the card on the input card listing.

The inputs are stored in data blocks that are held in a single large array. The blocks of each type are chained together by pointers to form a sequential file. For a detailed description of this system, see the description of common FILE (in Section 1.7), and then the internal data block descriptions (Section 1.8).

After all inputs have been read, they may be optionally re-printed in formatted tables. The purpose of this printout is to show all the inputs in an easily readable form.

The next processing performed is to generate a table of all user assigned names, and to replace all references to these names with internal ID numbers. Some of the subroutines that do this are table driven. That is, they locate where names are defined, and where they are referenced by the use of tables read from the initialization file. It is during this processing that undefined or multiply defined names are detected and flagged.

If no errors have been detected up to this point, the course block diagram is plotted (optional). Additional error messages are printed if the processing blocks are not in the proper order for plotting.

Next, a subroutine is called to complete the information in the CCTS, PMT group, and PMT data blocks. This consists of filling in the airbase number and the time, which are obtained from the airbase event that is specified by the block.

The final processing performed in phase 1 is to sort the input data and write it to file 10 for input to phase 2.

### Section 1.3

#### DESCRIPTION OF INPUTS

The input cards for the TRAM model have fixed format fields, and a separate card is provided for each different type of information. Coding forms for each of these input cards, along with a detailed description of the input variables, can be found in the Users' Guide. The following paragraphs give a more general description of the inputs.

All cards have a card name field, which is used to identify the card type. Although the different card types contain different information, most of them conform to a standard field layout. This consists of the card name field in columns 1-10, followed by two ten-column character data fields and ten five-column numeric data fields. All numeric data are read in integer format. Variables whose values can take on non-integral values are read with an implied decimal point. These values are converted internally, using the position of the implied decimal point shown on the coding forms. Character data are left justified and numeric data are right justified.

Cards that do not conform to the standard field layout (TASK, RUB, RUDB), must be preceded by a set header card. This card identifies the type of cards that are to follow. Note that these non-standard cards have a blank card name field, since the card type is identified by the header card. Each set is terminated by a SETEND card.

Some input cards require additional continuation cards. Such cards have parameters to give the number of each type of card that is to follow it. The formats for these additional cards are shown on the same coding form as the header card so that they may be easily coded in the proper sequence. Note that these continuation cards also have blank card name fields, since the card type is identified by the header card.

In general, these cards, or groups of cards, can be coded in any order. The only exception to this is the course and the processing block cards. The processing blocks for each course must follow the course card. Also, the processing blocks within a course must start with the graduation block and proceed



towards matriculation. This is because the position of each processing block is given as an offset from the block connected to it on the right. If there is more than one block to the right, as in the case of fan-outs, the first one encountered is the one used as the reference position.

The input deck must be ended with an end card, which consists of the word "END" punched in the first field.

## Section 1.4

### DESCRIPTION OF OUTPUTS

#### Input Card Listing

This listing shows the values exactly as they are read from each card. Field numbers are marked across the top of the page for reference by error messages. These fields correspond to five card columns. Character data are spread across two such fields, and would be referred to by the number of the second field. Also, card sequence numbers are printed for later reference by error messages. This listing is always printed.

#### Formatted Input Tables

These tables duplicate the information shown by the input card listing, but present it in a conveniently readable form. All variables are identified by column titles, and the meanings of integer codes are printed rather than the codes themselves. Also, these values will show the results of any input conversion that was done. The user may suppress this listing by use of the input routine control card.

#### Course Block Diagram

This plot shows the structure of the courses by displaying each processing block in the course as a rectangle, with the flow of students shown by connecting lines from one processing block to the next. Inside each rectangle, the block number, block name, synchronize-correlate reference (if any), and task names are shown. The course name is plotted above each graduation block. The input routine control card specifies if this plot is to be produced.



## Section 1.5

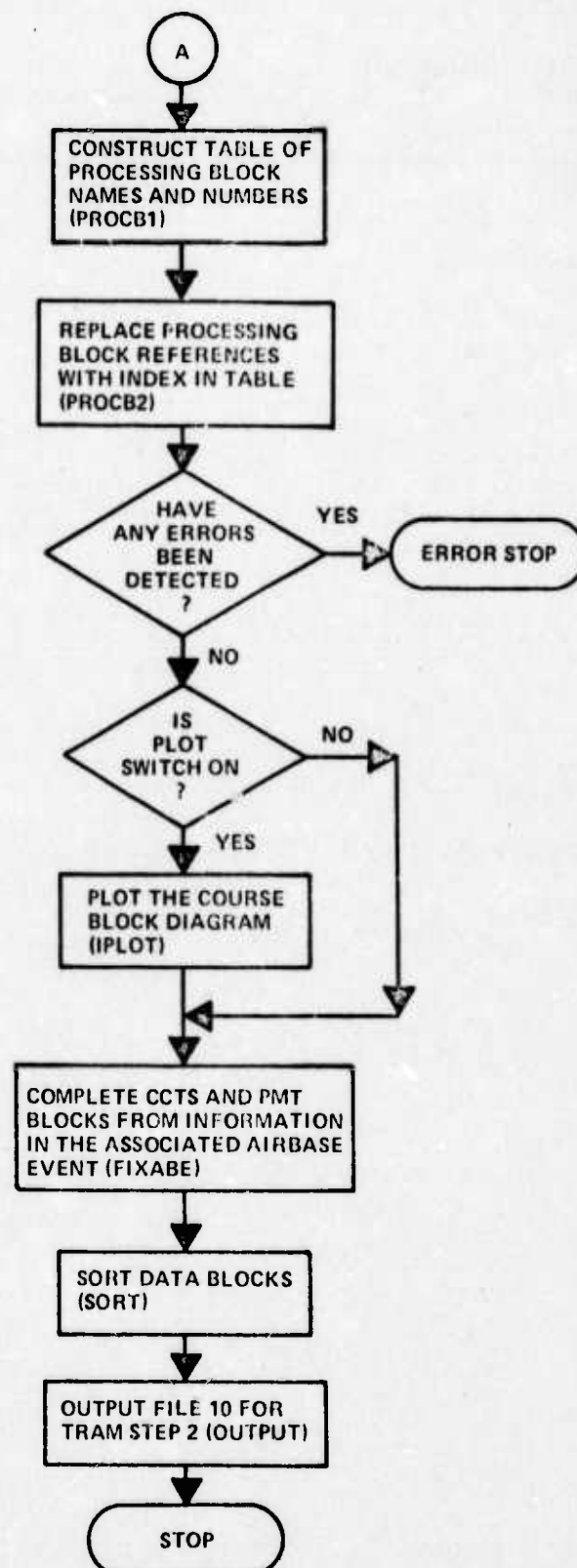
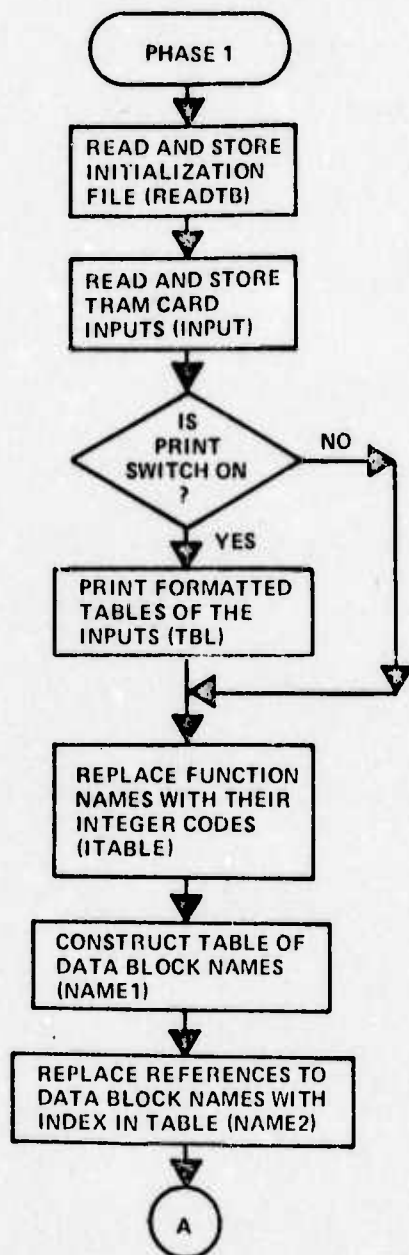
### SUBPROGRAM DESCRIPTIONS

This section contains the descriptions of the individual subroutines that comprise phase 1 of the TRAM program. The description for each subprogram consists of a statement of the purpose of the routine, the calling sequence, a description of its parameters, the method used, and a list of the subprograms required. A high level flowchart, which shows the logical decision points and the processing accomplished, is also included for each of the major subprograms.

```

CC***** PHASE1 *****
CC*
CC*          PHASE1 MAIN PROGRAM
CC*
CC*  PURPOSE
CC*    TO PERFORM THE FOLLOWING FUNCTIONS
CC*    - READ THE TRAM INPUT CARDS AND PRINT THEM
CC*    - CHECK VALUES FOR VALIDITY
CC*    - PRINT FORMATTED TABLES OF THE INPUTS
CC*    - REPLACE USER ASSIGNED NAMES WITH INTERNAL ID NUMBERS
CC*    - PLOT THE COURSE BLOCK DIAGRAM
CC*    - SORT THE INPUTS AND WRITE THEM TO UNIT 10 FOR PHASE 2
CC*
CC*  REFERENCES
CC*    SEE TRAM USERS GUIDE AND TRAM PROGRAMMERS MANUAL
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    READTB
CC*    INPUT
CC*    ITABLE
CC*    NAME1
CC*    NAME2
CC*    PROCB1
CC*    PROCB2
CC*    FIXABE
CC*    SORT
CC*    OUTPUT
CC*    IPLOT
CC*
CC*****

```



# PHASE 1 MAIN PROGRAM

```

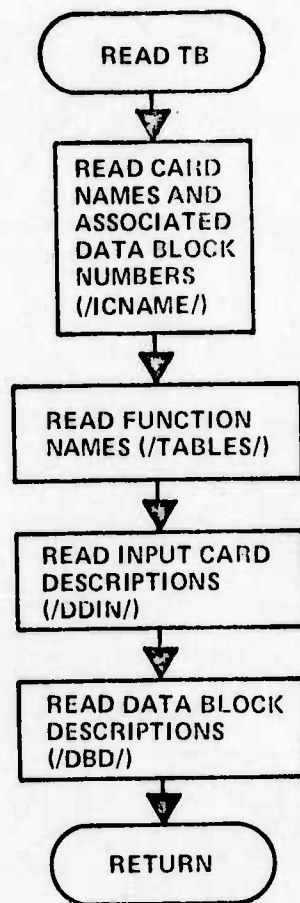
CC***** BLOCKD *****
CC*
CC*          BLOCK DATA BLOCKD
CC*
CC*  PURPOSE
CC*    TO INITIALIZE COMMON AREAS FOR INPUT ROUTINES
CC*
CC*****

```

```

CC***** READTB *****
CC*
CC*          SUBROUTINE READTB
CC*
CC*    PURPOSE
CC*      TO READ THE PHASE 1 INITIALIZATION FILE
CC*
CC*    CALLING SEQUENCE
CC*      CALL READTB
CC*
CC*    DESCRIPTION OF PARAMETERS
CC*      NONE
CC*
CC*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*      NONE
CC*****

```



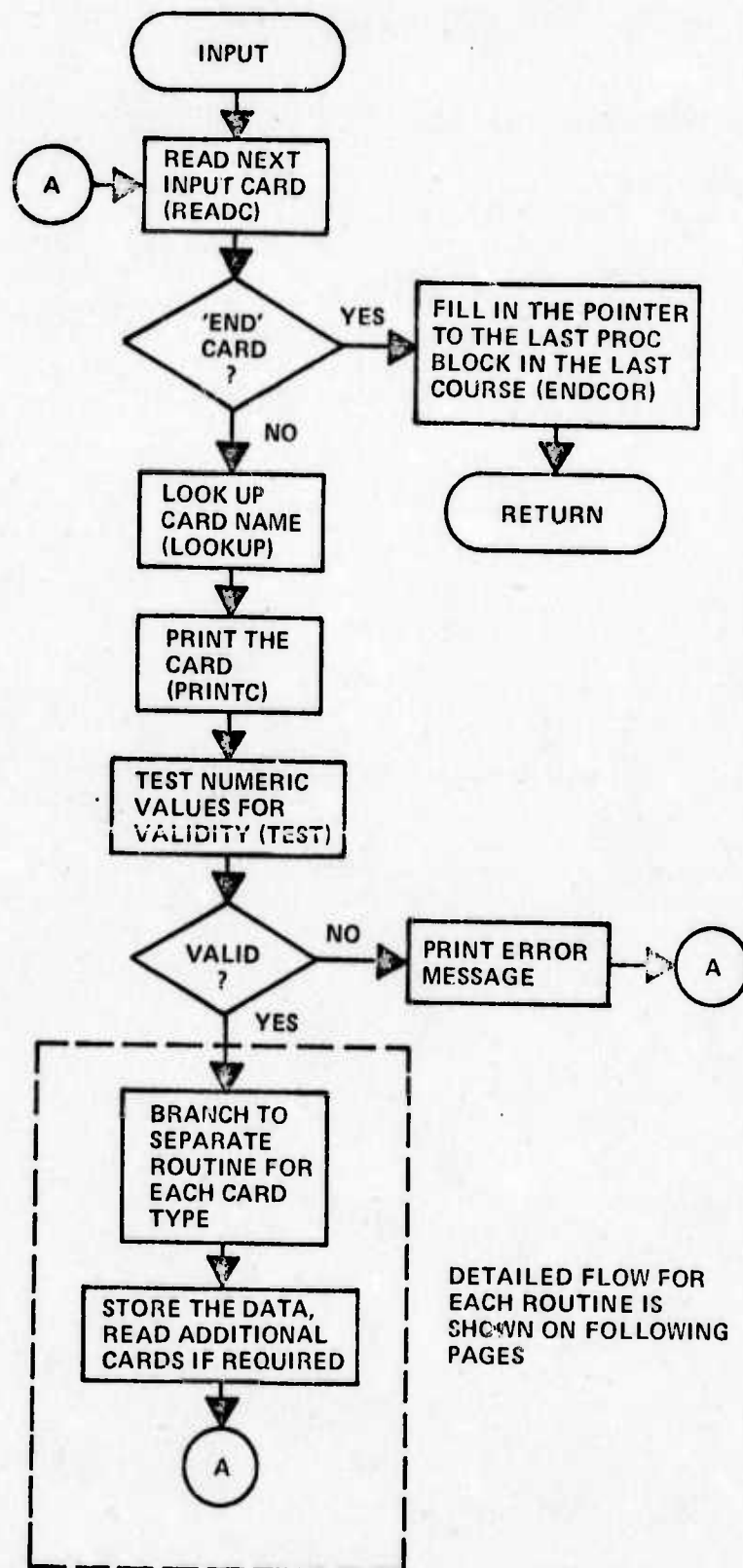
SUBROUTINE READTB

```

CC***** INPUT *****
CC*
CC*          SUBROUTINE INPUT
CC*
CC*    PURPOSE
CC*      TO READ THE INPUT CARDS AND STORE THEM
CC*
CC*    CALLING SEQUENCE
CC*      CALL INPUT (IERROR)
CC*
CC*    DESCRIPTION OF PARAMETERS
CC*      OUTPUT
CC*        IERROR - NUMBER OF ERRORS FOUND
CC*
CC*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*      READC
CC*      LOOKUP
CC*      PRINTC
CC*      TEST
CC*      RPT
CC*      STORE
CC*      SHIFTR
CC*      ENDCOR
CC*      ADDREC
CC*      TRANSFR
CC*
CC*****

```



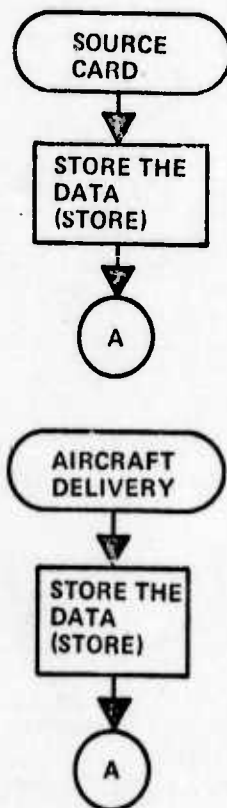


SUBROUTINE INPUT

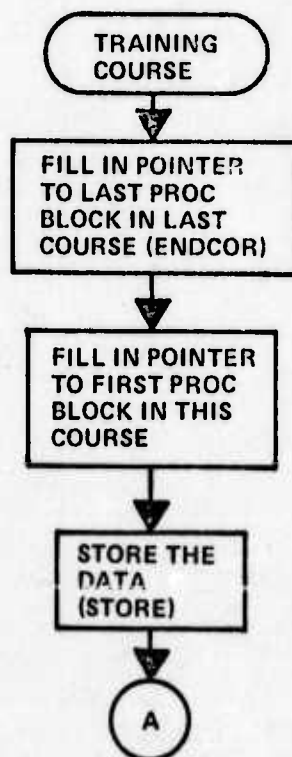




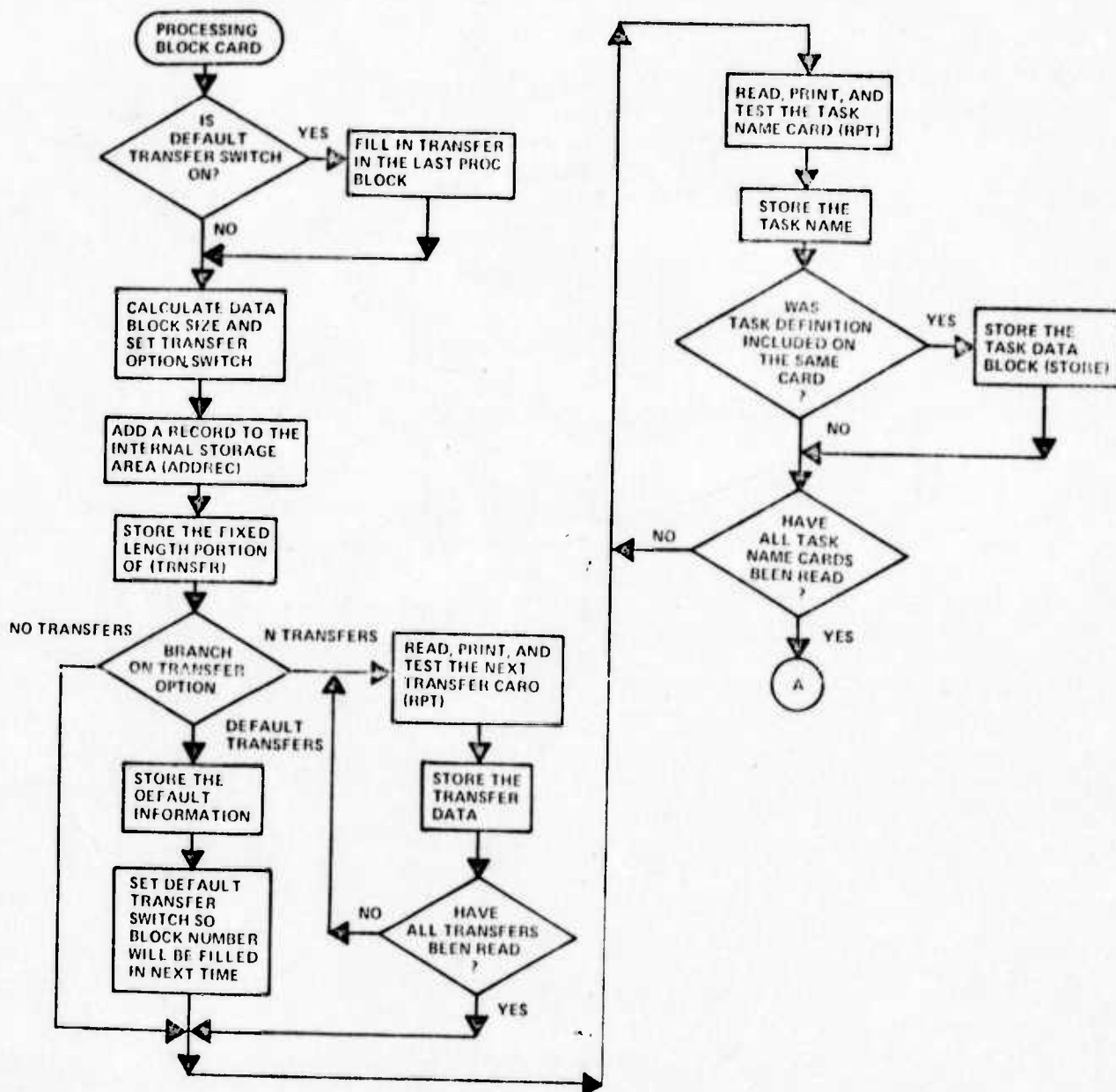
SUBROUTINE INPUT – CONTINUED



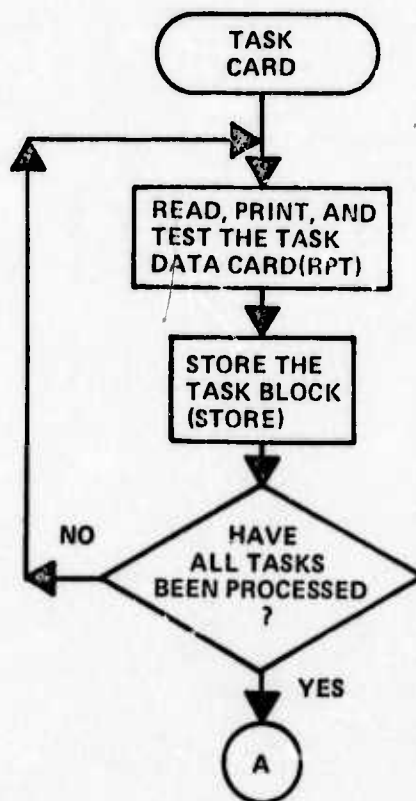
SUBROUTINE INPUT – CONTINUED



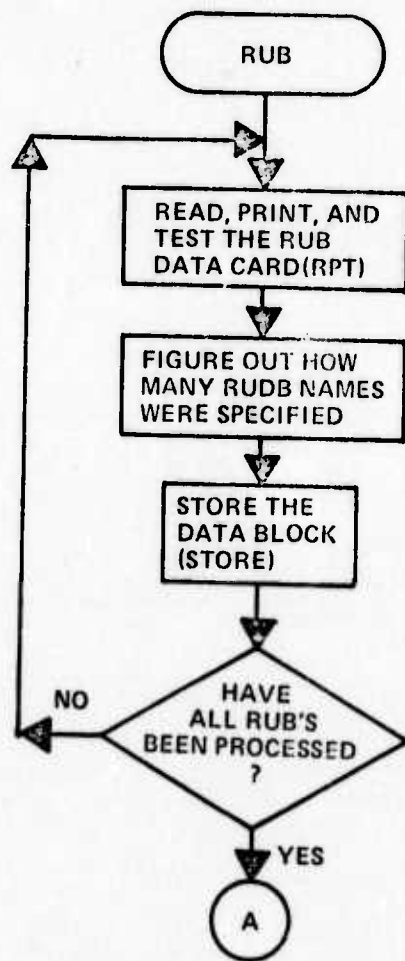
SUBROUTINE INPUT — CONTINUED



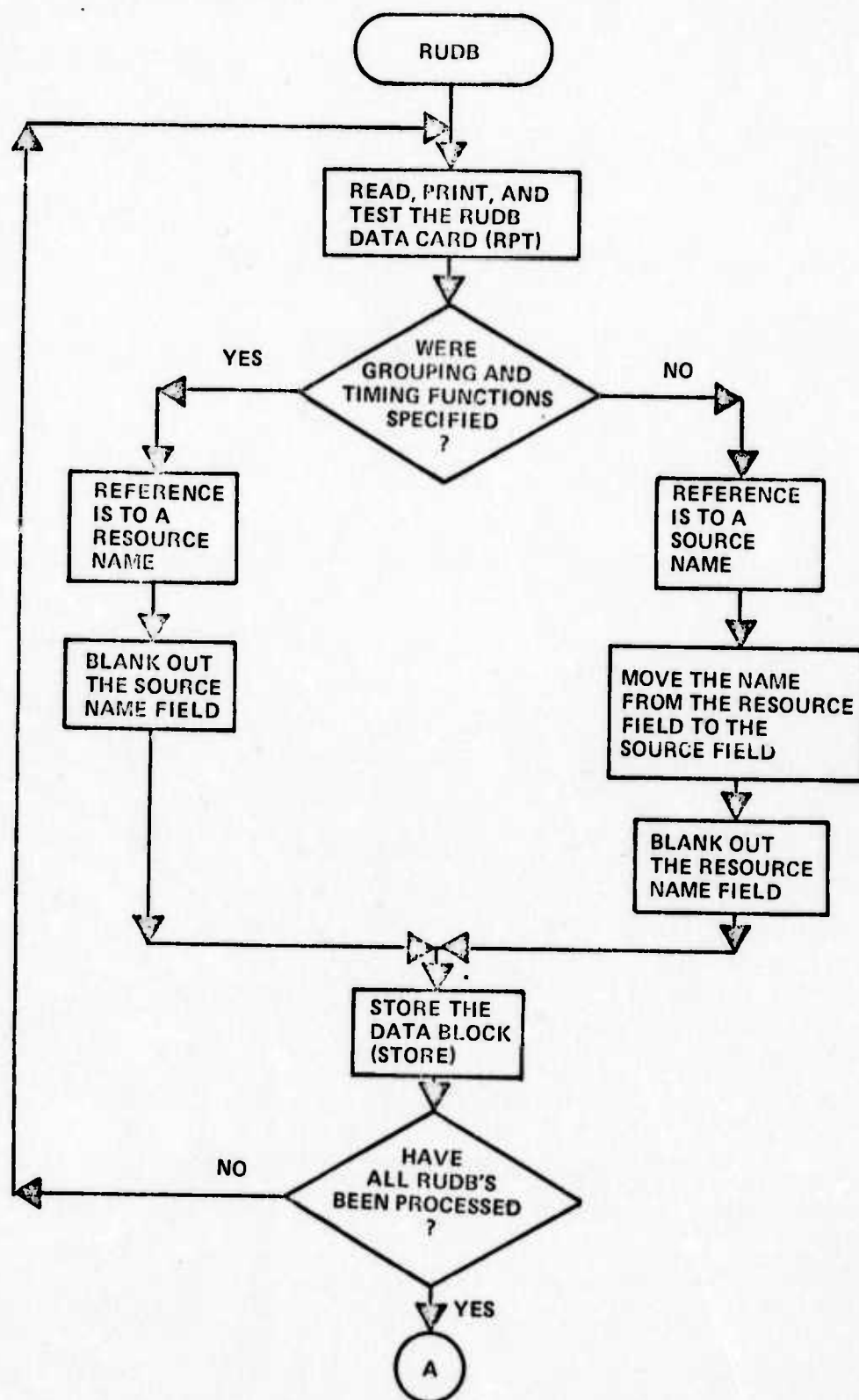
SUBROUTINE INPUT - CONTINUED



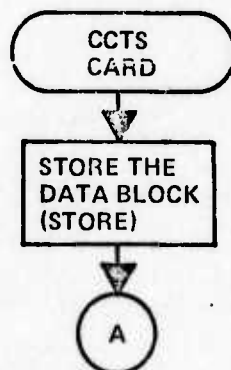
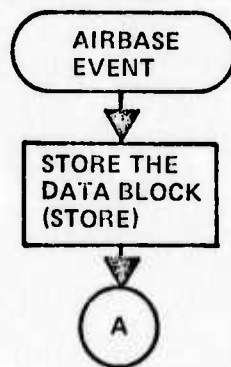
SUBROUTINE INPUT – CONTINUED



SUBROUTINE INPUT – CONTINUED

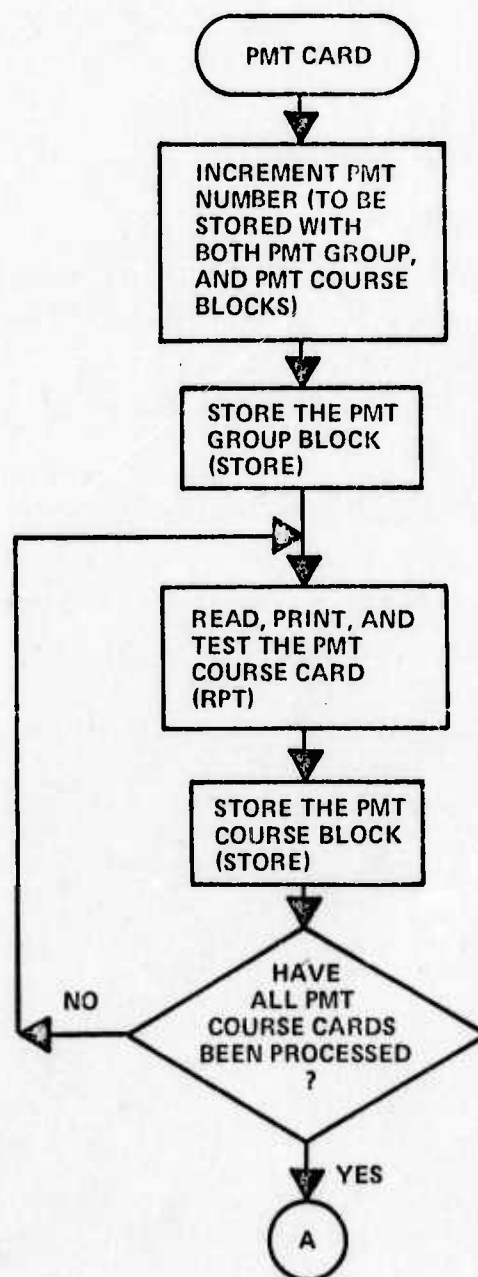


SUBROUTINE INPUT - CONTINUED



SUBROUTINE INPUT -- CONTINUED





SUBROUTINE INPUT - CONTINUED

```

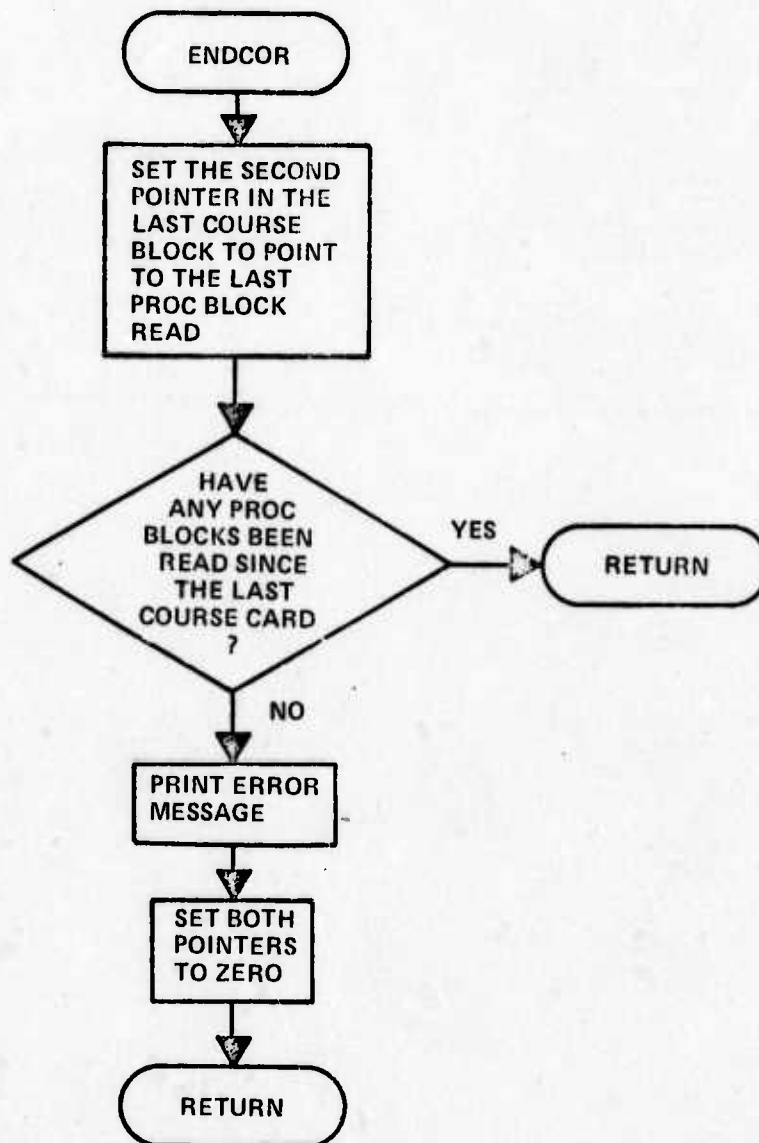
CC***** SHIFTR *****
CC*
CC*          SUBROUTINE SHIFTR
CC*
CC*  PURPOSE
CC*    TO REPACK THE ARRAY OF VALUES READ FROM INPUT CARDS IN ORDER
CC*    TO ELIMINATE WASTED STORAGE OCCUPIED BY UNUSED CHARACTER
CC*    FIELDS AT THE BEGINNING OF THE CARD
CC*
CC*  CALLING SEQUENCE
CC*    CALL SHIFTR (IP, N1, N2)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      N1    - NUMBER OF CHARACTER FIELDS WHICH ARE USED
CC*      N2    - NUMBER OF UNUSED CHARACTER FIELDS WHICH FOLLOW THE
CC*              FIRST N1 USED FIELDS
CC*    INPUT-OUTPUT
CC*      IP    - ARRAY OF PARAMETERS READ FROM A DATA CARD BY
CC*              SUBROUTINE READC
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** ENDCOR *****
CC*
CC*          SUBROUTINE ENDCOR
CC*
CC*  PURPOSE
CC*    TO SET THE PROCESSING BLOCK POINTERS IN THE TRAINING COURSE
CC*    BLOCK, AFTER ALL PROCESSING BLOCKS FOR THE COURSE HAVE BEEN
CC*    READ
CC*
CC*  CALLING SEQUENCE
CC*    CALL ENCOR (IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    OUTPUT
CC*      IERR - ERROR FLAG, SET WHEN COURSE CONTAINS NO PROCESSING
CC*           BLOCKS
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```



SUBROUTINE ENDCOR

```

CC***** READC *****
CC*
CC*          SUBROUTINE READC
CC*
CC*  PURPOSE
CC*    TO READ AN INPUT CARD
CC*
CC*  CALLING SEQUENCE
CC*    CALL READC (ICARD,IPARMS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICARD - CARD ID NUMBER OF CARD TO BE READ, OR
CC*              ZERO TO READ AN UNKNOWN CARD IN STANDARD FORMAT
CC*    OUTPUT
CC*      IPARMS- ARRAY OF UP TO 13 VALUES READ FROM THE CARD
CC*
CC*  REMARKS
CC*    THIS SUBROUTINE DETERMINES THE NUMBER OF PARAMETERS TO BE
CC*    READ, AND THEIR FORMAT FROM COMMON /DDIN/
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** PRINTC *****
CC*
CC*
CC*          SUBROUTINE PRINTC
CC*
CC*  PURPOSE
CC*    TO PRINT THE INPUT CARD LISTING
CC*
CC*  CALLING SEQUENCE
CC*    CALL PRINTC (ICARD,ISEQ,IPARMS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUTS
CC*      ICARD  - CARD NUMBER OF THE CARD TO BE PRINTED,
CC*              ZERO IF NOT KNOWN
CC*      ISEQ   - CARD SEQUENCE NUMBER
CC*      IPARMS - ARRAY OF VALUES THAT WERE READ FROM THE
CC*              CARD
CC*
CC*  REMARKS
CC*    1THIS SUBROUTINE USES COMMON /DDIN/ TO DETERMINE THE NUMBER
CC*      OF DATA FIELDS ON THE CARD, AND HOW MANY OF THEM ARE CHARACTER
CC*      DATA.
CC*    2 OUTPUT LINES ARE COUNTED, AND TITLES ARE PRINTED AT THE TOP
CC*      OF EACH PAGE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```



```

CC***** TEST *****
CC*
CC*          SUBROUTINE TEST
CC*
CC*  PURPOSE
CC*    TO TEST ALL NUMERIC INPUT VALUES TO SEE IF THEY ARE WITHIN
CC*    THE RANGE OF ACCEPTABLE VALUES
CC*
CC*  CALLING SEQUENCE
CC*    CALL TEST (ICARD, IARRAY, IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICARD - CARD NUMBER FROM WHICH THE VALUES WERE READ
CC*      IARRAY- ARRAY OF VALUES THAT WERE READ FROM THE
CC*              CARD
CC*    OUTPUT
CC*      IERR  - NUMBER OF ERRORS DETECTED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** STORE *****
CC*
CC*          SUBROUTINE STORE
CC*
CC*  PURPOSE
CC*    TO STORE A BLOCK OF INPUT DATA
CC*
CC*  CALLING SEQUENCE
CC*    CALL STURE (IBLK, NWDS, ISEQ, IP, IADDR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IBLK  - BLOCK NUMBER
CC*      NWDS  - NUMBER OF WORDS IN THE DATA BLOCK
CC*      ISEQ  - CARD SEQUENCE NUMBER TO BE ASSOCIATED WITH THIS BLOCK
CC*      IP    - ARRAY OF DATA WORDS TO BE STORED
CC*    OUTPUT
CC*      IADDR - POINTER TO LOCATION IN COMMON /IFILE/ WHERE THE DATA
CC*              WAS STORED (IF LESS THAN 1, NOT ENOUGH SPACE WAS
CC*              AVAILABLE IN /IFILE/ )
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    ADDREC
CC*    TRNSFR
CC*
CC*****

```

```

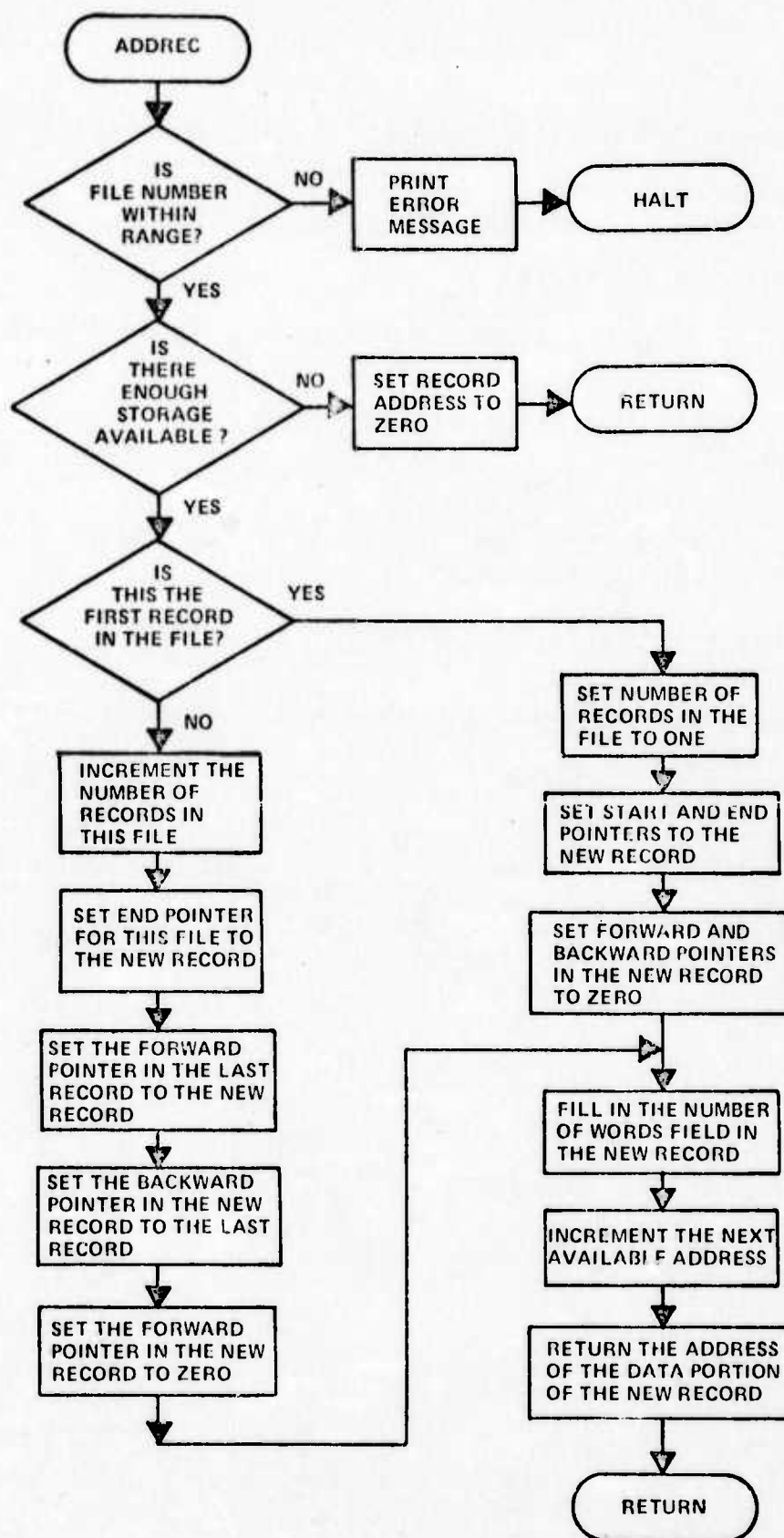
CC***** RPT *****
CC*
CC*          SUBROUTINE RPT
CC*
CC*  PURPOSE
CC*    TO READ AN INPUT CARD, PRINT IT, AND TEST THE NUMERIC FIELDS
CC*    FOR VALIDITY
CC*
CC*  CALLING SEQUENCE
CC*    CALL RPT (ICARD, ISEQ, IP, IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICARD  - CARD TYPE TO BE READ
CC*    INPUT-OUTPUT
CC*      ISEQ   - CARD SEQUENCE NUMBER, INCREMENTED BY EACH CALL
CC*    OUTPUT
CC*      IP     - ARRAY OF VALUES READ FROM THE CARD
CC*      IERR   - NUMBER OF ERRORS DETECTED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    READC
CC*    PRINTC
CC*    TEST
CC*****

```

```

CC***** ADDRESS *****
CC*
CC* SUBROUTINE ADDRESS *
CC*
CC* PURPOSE *
CC* TO ADD A RECORD TO A FILE IN COMMON AREA /IFILE/ *
CC*
CC* CALLING SEQUENCE *
CC* CALL ADDRESS (INDEX, NWDS, IADDR) *
CC*
CC* DESCRIPTION OF PARAMETERS *
CC* INPUT *
CC* INDEX - INDEX NUMBER OF THE FILE TO WHICH THE RECORD IS TO *
CC* ADDED *
CC* NWDS - NUMBER OF WORDS IN THE RECORD *
CC* OUTPUT *
CC* IADDR - ADDRESS OF THE DATA AREA OF THE NEW RECORD *
CC* (SUBSCRIPT IN IFILE ARRAY) SET TO ZERO IF ERROR *
CC* OCCURS *
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED *
CC* NONE *
CC*****

```



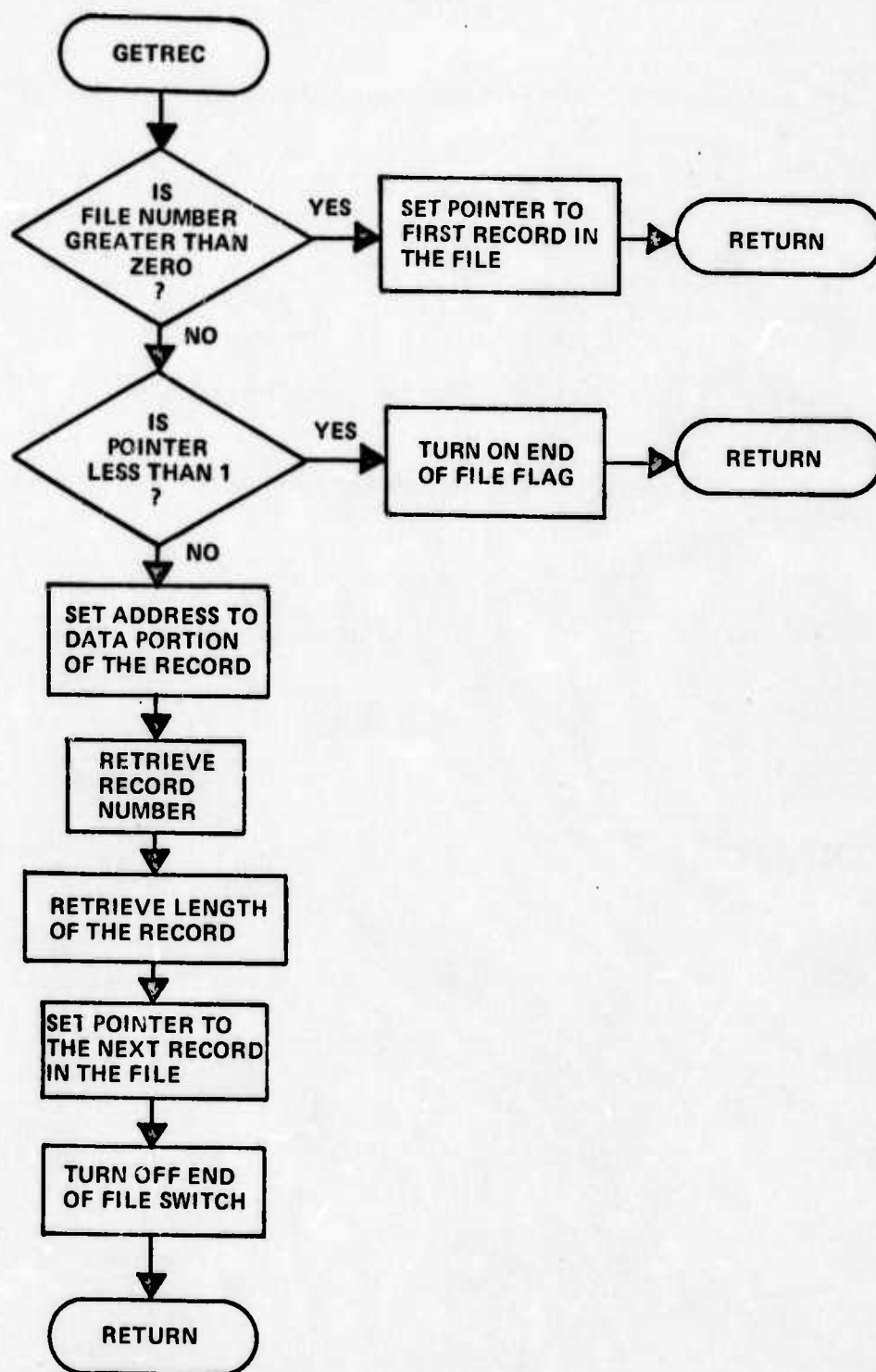
SUBROUTINE ADDREC

```

CC***** GETREC *****
CC*
CC*          SUBROUTINE GETREC
CC*
CC*  PURPOSE
CC*    TO LOCATE THE NEXT SEQUENTIAL RECORD IN A FILE
CC*
CC*  CALLING SEQUENCE
CC*    CALL GETREC (INDEX,IREC,NWDS,IADDR,IEOF)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      INDEX - FILE INDEX NUMBER FOR INITIALIZATION CALL
CC*              OR
CC*      ZERO TO LOCATE THE NEXT RECORD IN THE FILE SPECIFIED
CC*      IN THE LAST INITIALIZATION CALL
CC*    OUTPUT
CC*      IREC - RECORD NUMBER
CC*      NWDS - NUMBER OF WORDS IN THE RECORD
CC*      IADDR - ADDRESS (SUBSCRIPT IN IFILE ARRAY) OF DATA
CC*      IEOF - END OF FILE FLAG
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```





SUBROUTINE GETREC

```

CC***** TRANSFER *****
CC*
CC*                                SUBROUTINE TRANSFER
CC*
CC*    PURPOSE
CC*        TO TRANSFER AN ARRAY FROM ONE LOCATION TO ANOTHER
CC*
CC*    CALLING SEQUENCE
CC*        CALL TRANSFER (I1, I2, NWDS)
CC*
CC*    DESCRIPTION OF PARAMETERS
CC*        I1      - ARRAY TO BE MOVED
CC*        I2      - ARRAY TO WHICH I1 IS TO BE MOVED
CC*        NWDS    - NUMBER OF ELEMENTS TO BE MOVED
CC*
CC*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*        NONE
CC*
CC*****

```

```

CC***** TBL *****
CC*
CC*          SUBROUTINE TBL
CC*
CC*  PURPOSE
CC*    TO PRINT THE FORMATTED TABLES OF THE INPUTS
CC*
CC*  CALLING SEQUENCE
CC*    CALL TBL
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  METHOD
CC*    THIS SUBROUTINE CALLS A SEPARATE SUBROUTINE TO PRINT EACH OF
CC*    THE DATA TABLES
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    TBL2
CC*    TBL3
CC*    TBL4
CC*    TBL5
CC*    TBL6
CC*    TBL7
CC*    TBL8
CC*    TBL9
CC*    TBL10
CC*    TBL12
CC*    TBL13
CC*    TBL14
CC*
CC*****

```

```

CC***** TBL2 *****
CC*
CC*          SUBROUTINE TBL2
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF AIRBASE INVENTORIES
CC*
CC*****

```

```

CC***** TBL3 *****
CC*
CC*          SUBROUTINE TBL3
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF THE RESOURCE INVENTORIES
CC*
CC*****

```

```

CC***** TBL4 *****
CC*
CC*          SUBROUTINE TBL4
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF THE SOURCE CARDS
CC*
CC*****

```

```

CC***** TBL5 *****
CC*
CC*          SUBROUTINE TBL5
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF AIRCRAFT DELIVERY INPUTS
CC*
CC*****

```

```

CC***** TBL6 *****
CC*
CC*          SUBROUTINE TBL6
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF COURSE BLOCKS
CC*
CC*****

```

```

CC***** TBL7 *****
CC*
CC*          SUBROUTINE TBL7
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF THE PROCESSING BLOCKS
CC*
CC*****

```

```

CC***** TBL8 *****
CC*
CC*          SUBROUTINE TBL8
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF TASK BLOCKS
CC*
CC*****

```

```

CC***** TBL9 *****
CC*
CC*          SUBROUTINE TBL9
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF RESOURCE UTILIZATION BLOCKS
CC*
CC*****

```

```

CC***** TBL10 *****
CC*
CC*          SUBROUTINE TBL10
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF RESOURCE UTILIZATION DESCRIPTION BLOCKS
CC*
CC*****

```

```

CC***** TBL12 *****
CC*
CC*          SUBROUTINE TBL12
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF AIRBASE EVENT CARDS
CC*
CC*****

```

```

CC***** TBL13 *****
CC*
CC*          SUBROUTINE TBL13
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF CCTS CARDS
CC*
CC*****

```

```

CC***** TBL14 *****
CC*
CC*          SUBROUTINE TBL14
CC*
CC*  PURPOSE
CC*    TO PRINT THE TABLE OF PMT CARDS
CC*
CC*****

```



```

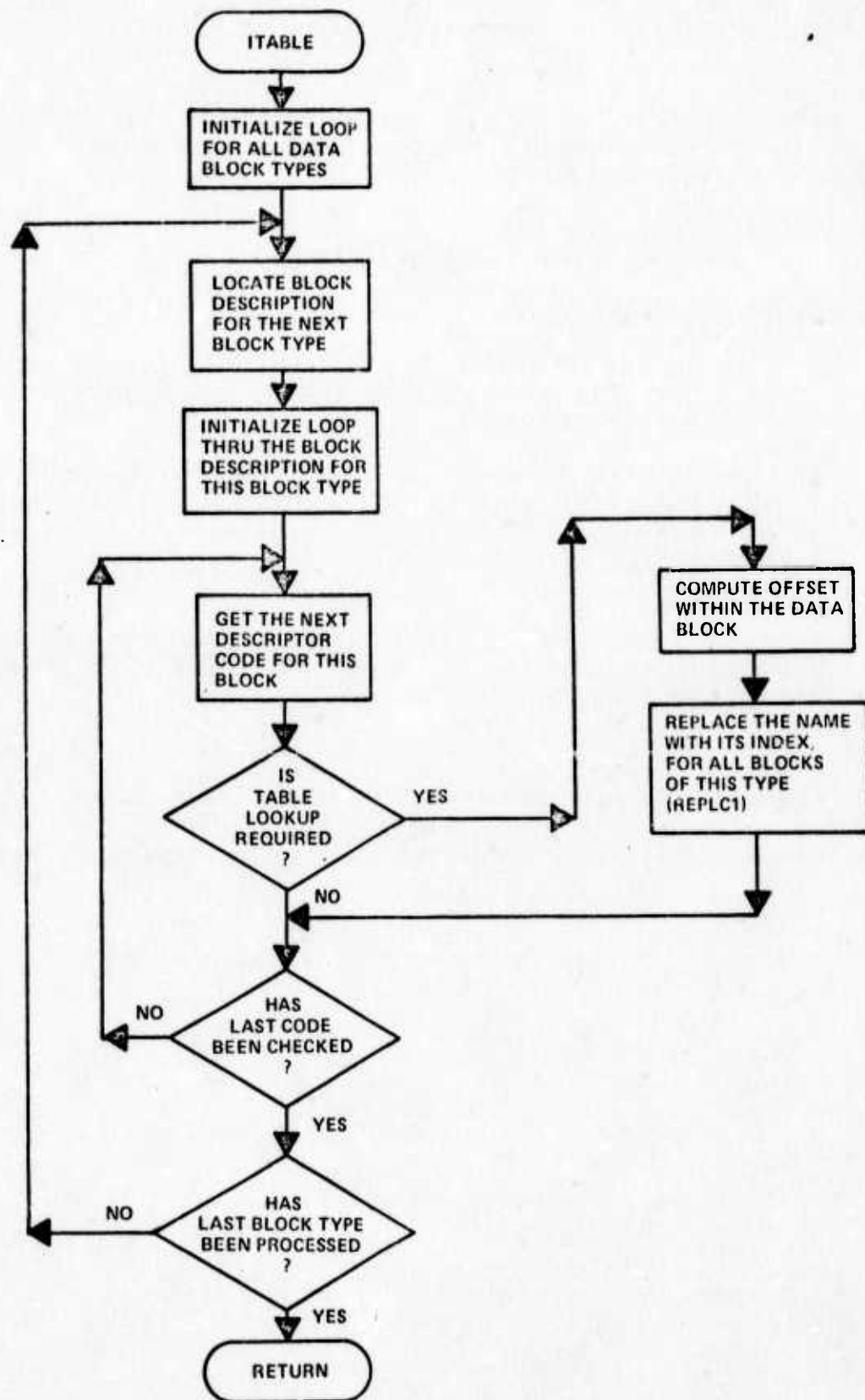
CC***** PTYPE *****
CC*
CC*          SUBROUTINE PTYPE
CC*
CC*  PURPOSE
CC*    TO DECODE THE PERSONNEL TYPE CODE AND RETURN A CHARACTER
CC*    ARRAY OF THE TYPES SPECIFIED BY THE CODE. THE PERSONNEL
CC*    TYPE CODE IS A 4 BIT INTEGER NUMBER, WITH EACH BIT USED
CC*    TO INDICATE A PERSONNEL TYPE AS FOLLOWS (GOING FROM
CC*    LEFT TO RIGHT)
CC*      1  PILOTS
CC*      2  CGPILOTS
CC*      3  DSO
CC*      4  DSO
CC*
CC*  CALLING SEQUENCE
CC*    CALL PTYPE (ICODE, IARRAY)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICODE - PERSONNEL TYPE CODE
CC*    OUTPUT
CC*      IARRAY - ARRAY OF PERSONNEL TYPE NAMES SPECIFIED BY ICODE
CC*
CC*****

```

```

CC***** ITABLE *****
CC*
CC*          SUBROUTINE ITABLE
CC*
CC*  PURPOSE
CC*    TO REPLACE CHARACTER INPUT PARAMETERS WITH ITS INTEGER CODE.
CC*    ERROR MESSAGES ARE PRINTED FOR VALUES NOT FOUND IN THE TABLES.
CC*
CC*  CALLING SEQUENCE
CC*    CALL ITABLE (IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    OUTPUT
CC*    IERR  - NUMBER OF ERRORS ENCOUNTERED
CC*
CC*  METHOD
CC*    THIS SUBROUTINE USES COMMON /DBD/ TO LOCATE THESE PARAMETERS
CC*    AND TO FIND OUT WHICH TABLE THEY ARE TO BE LOOKED UP IN
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    REPLC1
CC*
CC*****

```

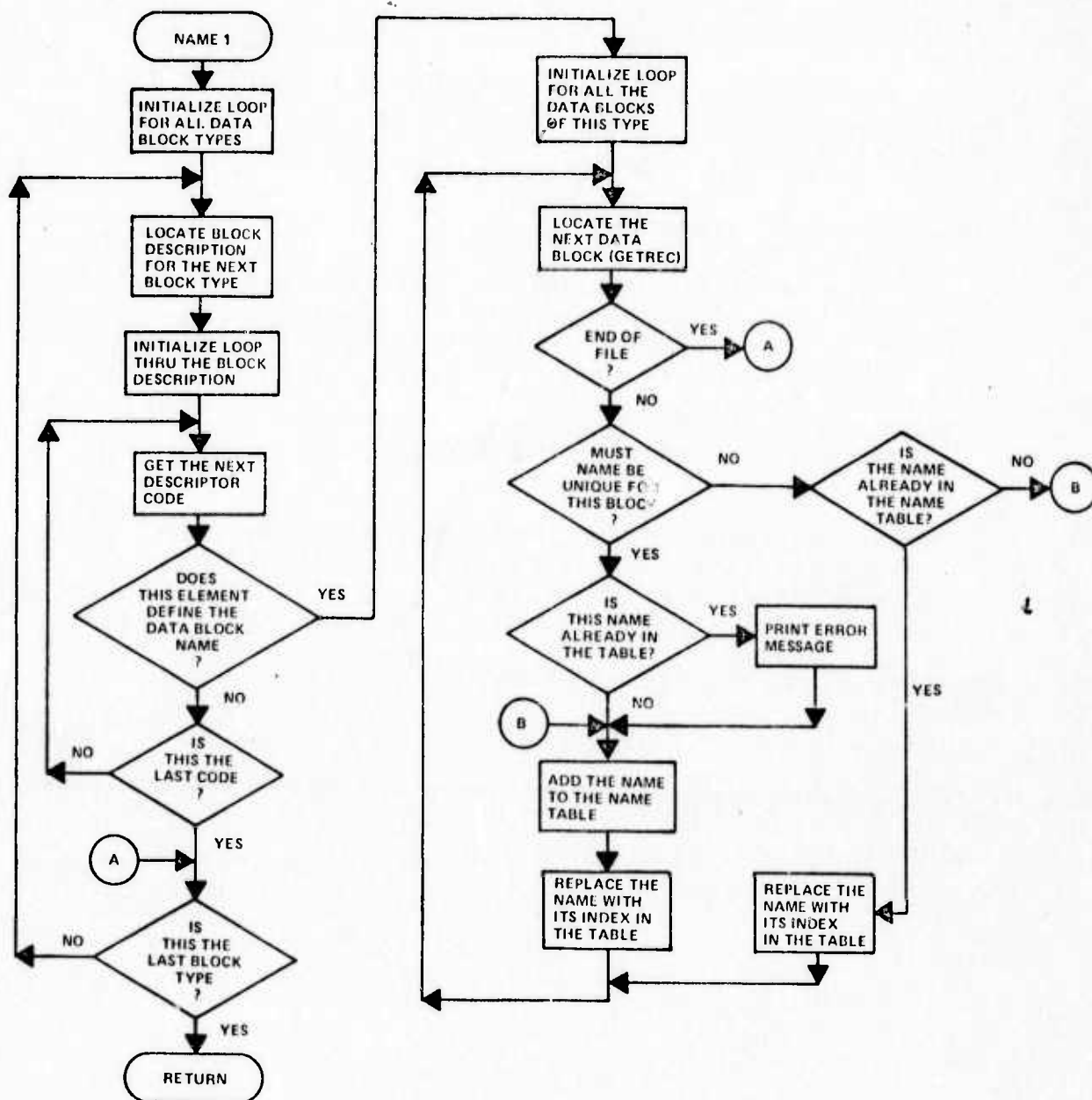


SUBROUTINE ITABLE

```

CC***** NAME1 *****
CC*
CC*          SUBROUTINE NAME1
CC*
CC*  PURPOSE
CC*    TO MAKE A TABLE OF THE NAMES OF ALL NAMED DATA BLOCKS, AND TO
CC*    REPLACE THE CHARACTER NAME IN THE DATA BLOCK WITH THE INDEX
CC*    OF THE NAME IN THE TABLE
CC*
CC*  CALLING SEQUENCE
CC*    CALL NAME1 (IP,INUM,ITBL,NTBL,IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT-OUTPUT
CC*      NTBL  - NUMBER OF ELEMENTS IN ITBL ARRAY (GIVES NUMBER OF
CC*             ELEMENTS AVAILABLE AT ENTRY, AND NUMBER OF ELEMENTS
CC*             USED AT EXIT)
CC*    OUTPUT
CC*      IP    - POINTER TO START OF NAME TABLE FOR EACH DATA BLOCK
CC*      INUM  - NUMBER OF ENTRIES IN NAME TABLE FOR EACH DATA BLOCK
CC*      ITBL  - NAME TABLES (CONTAINS ORIGINAL CHARACTER NAMES)
CC*      IERR  - NUMBER OF ERRORS ENCOUNTERED
CC*
CC*  METHOD
CC*    THIS SUBROUTINE USES COMMON /DBD/ TO DETERMINE WHICH DATA
CC*    BLOCKS ARE NAMED, AND TO LOCATE THE POSITION OF THE NAME
CC*    WITHIN THE BLOCK
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*    LOOKUP
CC*
CC*****

```



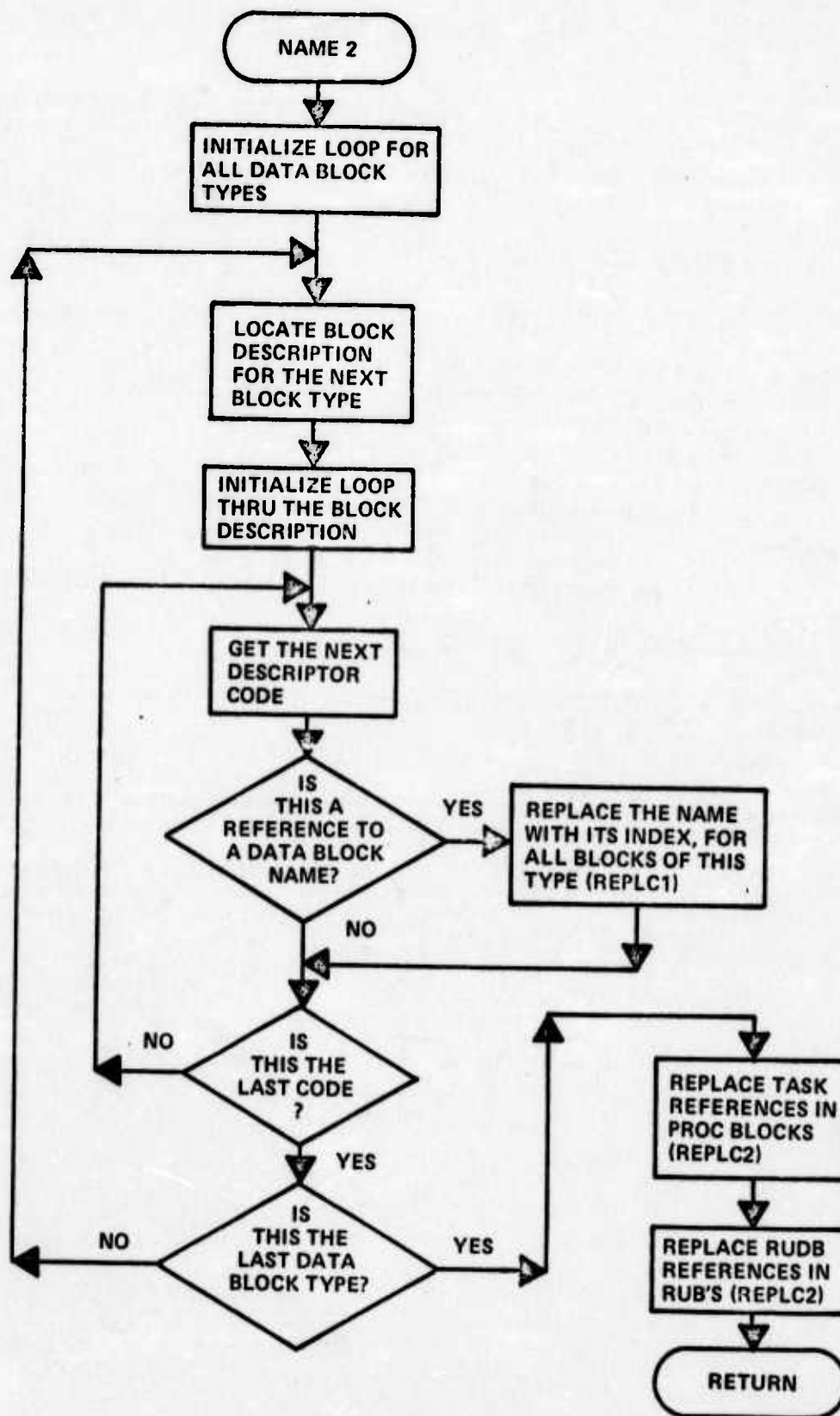
SUBROUTINE NAME1

```

CC***** NAME2 *****
CC*
CC*          SUBROUTINE NAME2
CC*
CC*  PURPOSE
CC*    TO REPLACE ALL BLOCK NAME REFERENCES WITH THEIR INTEGER CODES
CC*
CC*  CALLING SEQUENCE
CC*    CALL NAME2 (IP,INUM,ITBL,IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IP      - ARRAY OF POINTERS TO FIRST ELEMENT OF NAME TABLE
CC*                FOR EACH DATA BLOCK
CC*      INUM    - ARRAY GIVING NUMBER OF ENTRIES IN NAME TABLE FOR
CC*                EACH DATA BLOCK
CC*      ITBL    - NAME TABLE
CC*    OUTPUT
CC*      IERR    - NUMBER OF ERRORS DETECTED
CC*
CC*  METHOD
CC*    THIS SUBROUTINE USES COMMON /DBD/ TO LOCATE BLOCK NAME
CC*    REFERENCES IN THE FIXED PORTION OF DATA BLOCKS. THE POSITION*
CC*    OF NAME REFERENCES IN VARIABLE LENGTH PORTION OF DATA BLOCKS *
CC*    IS HARD CODED (THIS OCCURS IN PROCBLOCK AND RUB)
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    REPLC1
CC*    REPLC2
CC*
CC*****

```



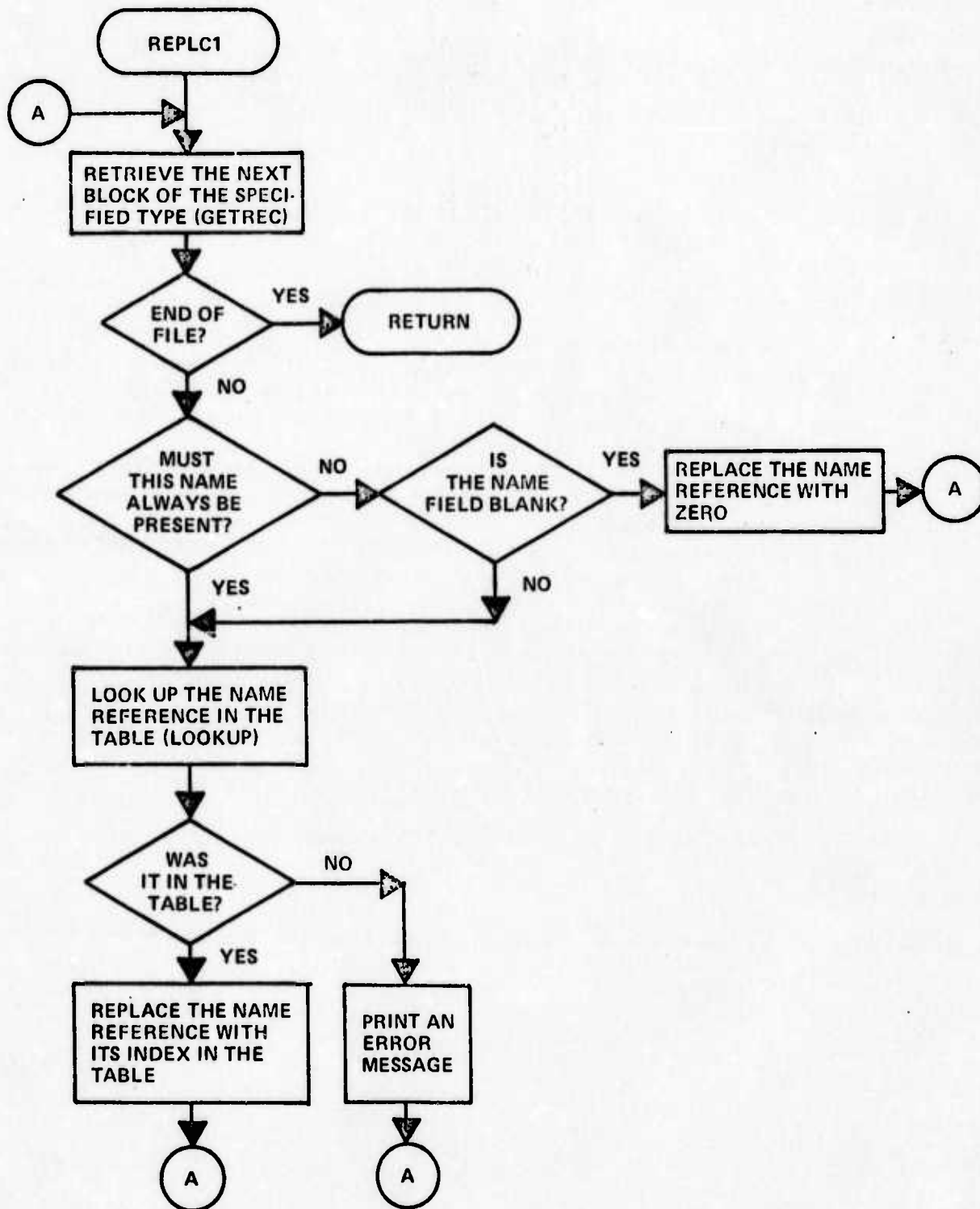


SUBROUTINE NAME2

```

CC***** REPLC1 *****
CC*
CC* SUBROUTINE REPLC1
CC*
CC* PURPOSE
CC* TO REPLACE A BLOCK NAME REFERENCE WITH ITS INTEGER CODE IN
CC* ALL DATA BLOCKS OF A GIVEN TYPE. THIS ROUTINE IS ONLY FOR
CC* REFERENCES IN THE FIXED PORTION OF VARIABLE LENGTH DATA BLOCKS*
CC*
CC* CALLING SEQUENCE
CC* CALL REPLC1 (IBLK, IOFF, ITBL, NTBL, INDOFF, ICODE, IERR)
CC*
CC* DESCRIPTION OF PARAMETERS
CC* INPUT
CC* IBLK - DATA BLOCK NUMBER
CC* IOFF - OFFSET OF REFERENCE IN THE DATA BLOCK
CC* ITBL - NAME TABLE
CC* NTBL - NUMBER OF ELEMENTS IN ITBL
CC* INDOFF- OFFSET TO BE ADDED TO THE INDEX OF A NAME IN THE
CC* TABLE IN ORDER TO GET ITS INTEGER CODE
CC* ICODE - 1 IF THE NAMES ARE INTEGER DATA
CC* 2 IF THE NAMES ARE CHARACTER DATA
CC* 3 IF CHARACTER DATA AND BLANKS ARE ALLOWED
CC*
CC* OUTPUT
CC* IERR - NUMBER OF ERRORS DETECTED
CC*
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC* GETREC
CC* LOOKUP
CC*
CC*****

```

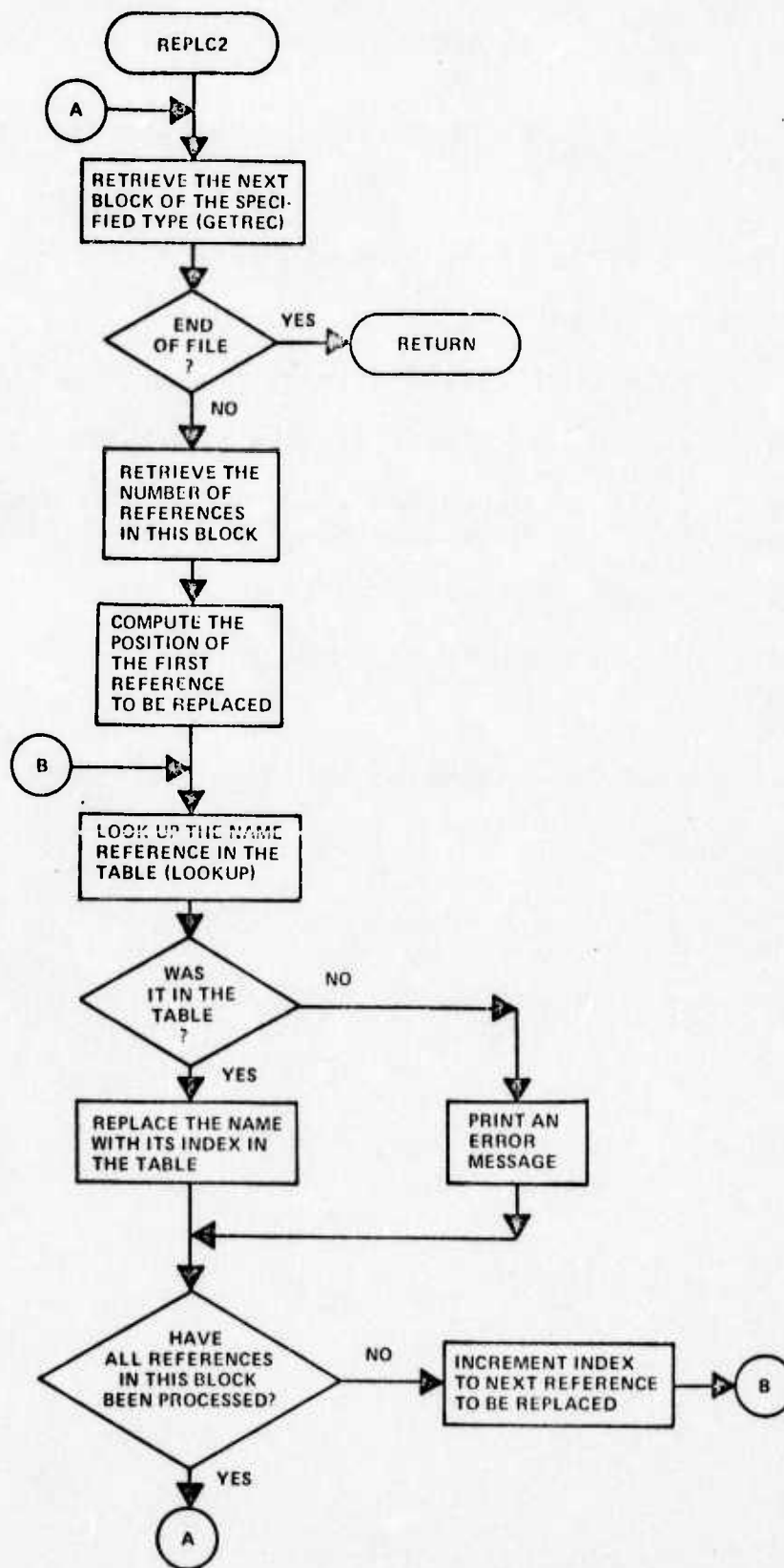


SUBROUTINE REPLC1

```

CC***** REPLC2 *****
CC*
CC*          SUBROUTINE REPLC2
CC*
CC*  PURPOSE
CC*    TO REPLACE A BLOCK NAME REFERENCE WITH ITS INTEGER CODE IN
CC*    ALL DATA BLOCKS OF A GIVEN TYPE. THIS ROUTINE IS ONLY FOR
CC*    REFERENCES IN THE VARIABLE PORTION OF VARIABLE LENGTH DATA
CC*    BLOCKS
CC*
CC*  CALLING SEQUENCE
CC*    CALL REPLC2 (IBLK,IOFF1,IDX,NDX,N,IOFF2,ITBL,NTBL,INDOFF,
CC*               ICODE, IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IBLK - DATA BLOCK NUMBER
CC*      IOFF1 - OFFSET WITHIN THE DATA BLOCK TO THE WORDS GIVING THE
CC*              NUMBER OF ENTRIES IN EACH VARIABLE LENGTH ITEM
CC*      IDX1 - ARRAY WHICH GIVES THE NUMBER OF WORDS IN EACH ENTRY
CC*              OF EACH VARIABLE LENGTH ITEM
CC*      NDX - NUMBER OF VARIABLE LENGTH ITEMS IN THE DATA BLOCK
CC*      N - THE NUMBER OF THE ITEM IN WHICH THE REPLACEMENT IS
CC*              TO BE DONE
CC*      IOFF2 - THE OFFSET WITHIN THE ITEM OF THE WORD CONTAINING
CC*              THE REFERENCES TO BE REPLACED
CC*      ITBL - NAME TABLE
CC*      NTBL - NUMBER OF NAMES IN ITBL
CC*      INDOFF- OFFSET TO BE ADDED TO A NAMES INDEX IN THE NAME TABLE*
CC*              TO GET ITS INTEGER CODE
CC*      ICODE - 2 IF NAMES ARE CHARACTER DATA
CC*              1 IF NAMES ARE INTEGER DATA
CC*    OUTPUT
CC*      IERR - NUMBER OF ERRORS DETECTED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*    LOOKUP
CC*****

```



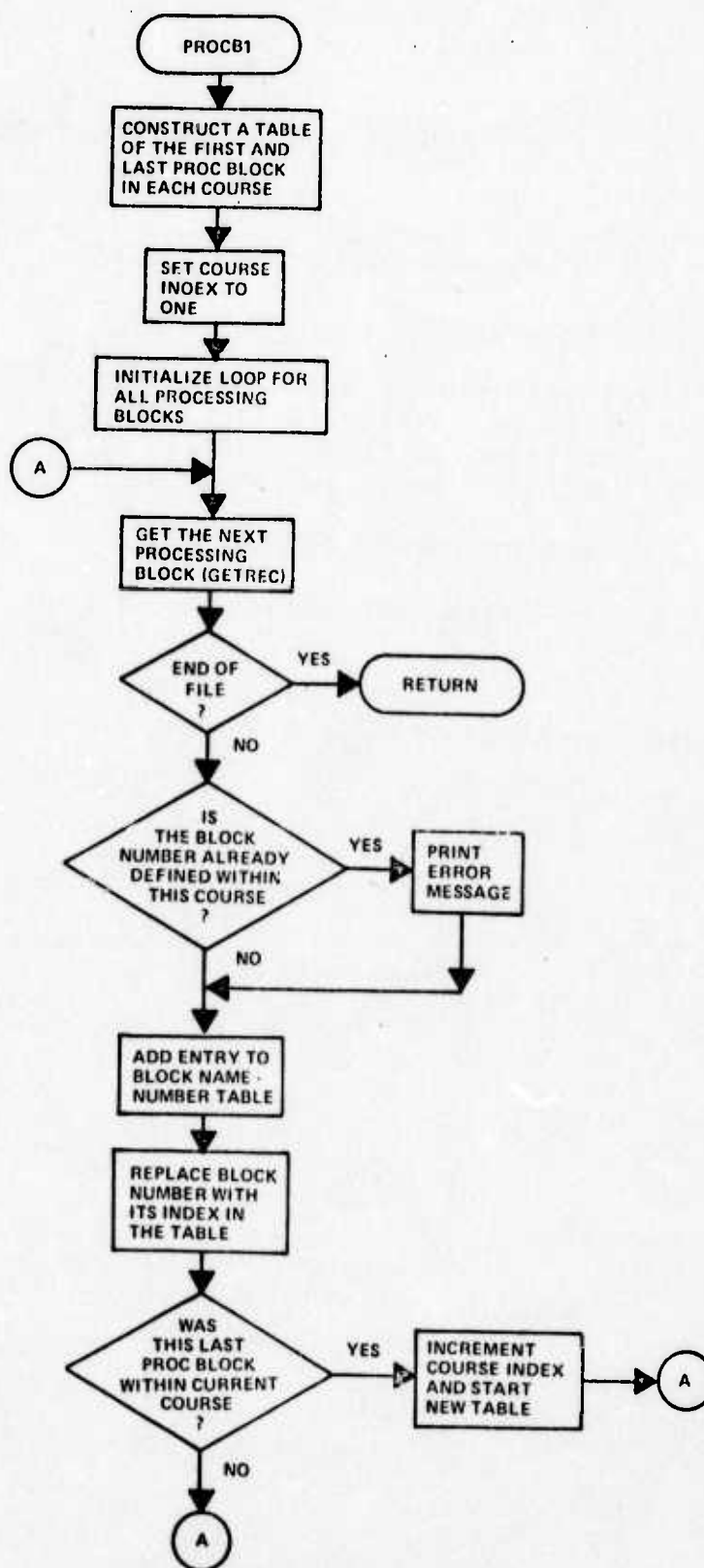
SUBROUTINE REPLC2

```

CC***** PROCB1 *****
CC*
CC*          SUBROUTINE PROCB1
CC*
CC*  PURPOSE
CC*    TO CONSTRUCT A TABLE OF PROCESSING BLOCK NAMES AND NUMBERS
CC*    ASSIGNED BY THE USER
CC*
CC*  CALLING SEQUENCE
CC*    CALL PROCB1 (IP1, INUM, NAME, NUMBER, IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    OUTPUT
CC*      IP1   - POINTER TO START OF NAME AND NUMBER TABLES FOR EACH
CC*             COURSE
CC*      INUM  - NUMBER OF ENTRIES IN NAME AND NUMBER TABLES FOR
CC*             EACH COURSE
CC*      NAME  - TABLE OF NAMES FOR EACH PROCESSING BLOCK
CC*      NUMBER- TABLE OF USER ASSIGNED NUMBERS FOR EACH PROCESSING
CC*             BLOCK
CC*      IERR  - NUMBER OF ERRORS DETECTED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*    LOOKUP
CC*
CC*****

```



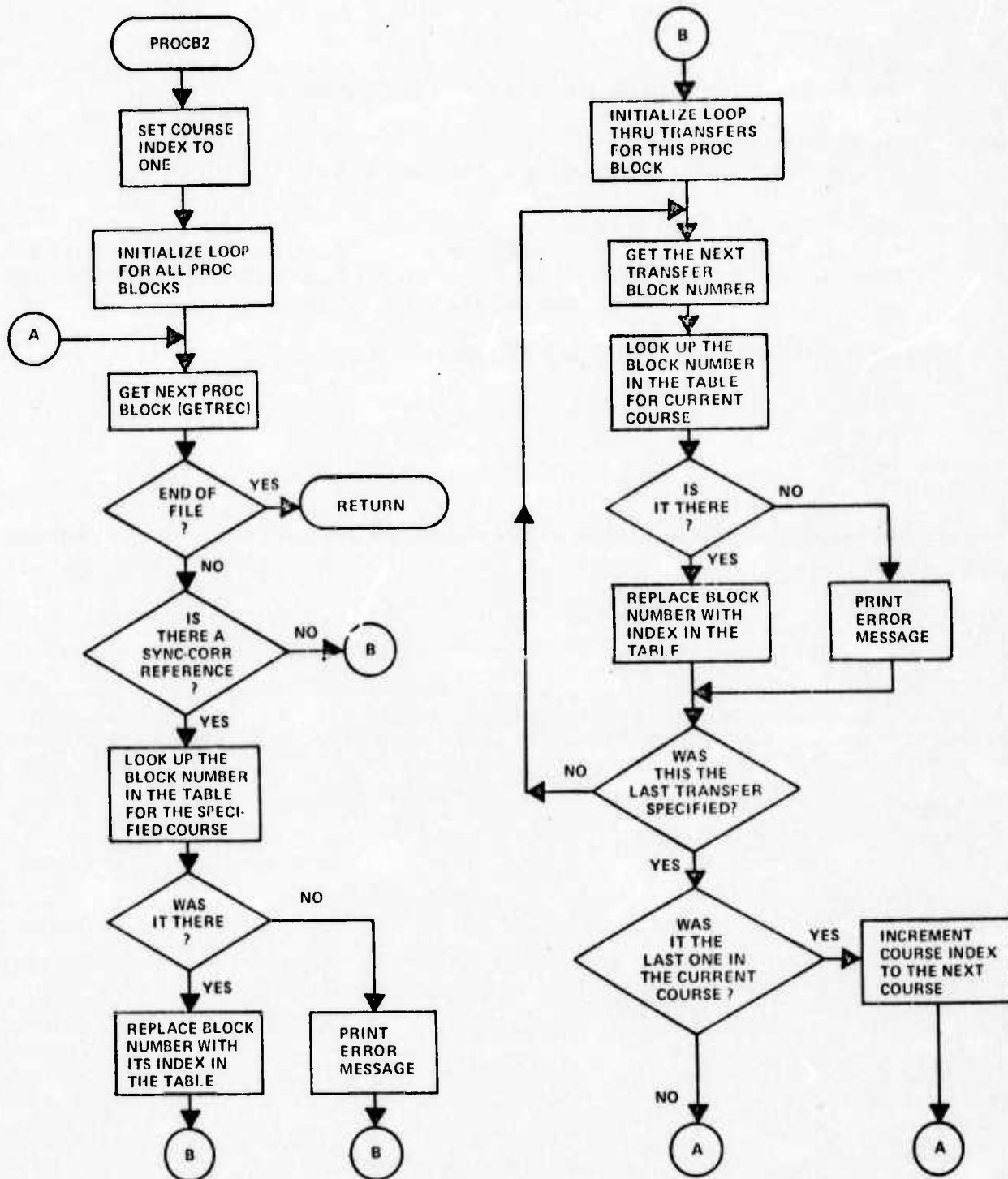


SUBROUTINE PROCB1

```

CC***** PRCCB2 *****
CC*
CC*          SUBROUTINE PROCB2
CC*
CC*  PURPOSE
CC*    TO REPLACE ALL BLOCK NUMBER REFERENCES WITH THEIR INTERNAL
CC*    ID NUMBERS
CC*
CC*  CALLING SEQUENCE
CC*    CALL PROCB2 (IPI,INUM,NAME,NUMBER,IERR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IPI   - POINTER TO NAME AND NUMBER TABLE FOR EACH COURSE
CC*      INUM  - NUMBER OF ENTRIES IN TABLE FOR EACH COURSE
CC*      NAME  - PROCESSING BLOCK NAME TABLE
CC*      NUMBER- PROCESSING BLOCK NUMBER TABLE
CC*    OUTPUT
CC*      IERR  - NUMBER OF ERRORS DETECTED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*    LOOKUP
CC*****

```

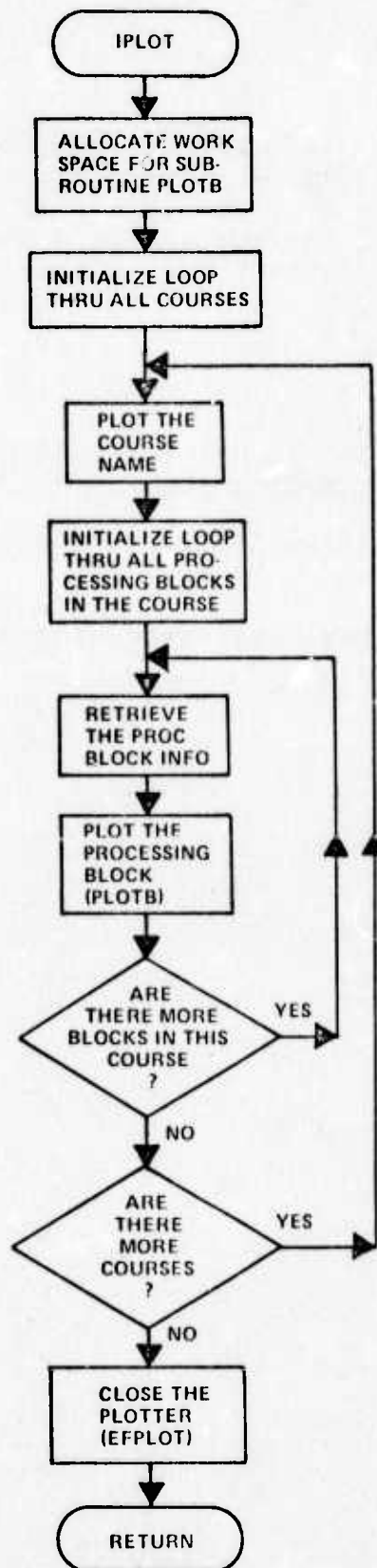


SUBROUTINE PROCB2

```

CC***** I P L O T *****
CC*
CC*          SUBROUTINE IPLOT
CC*
CC*  PURPOSE
CC*    TO PRODUCE THE CALCOMP PLOT OF THE COURSES
CC*
CC*  CALLING SEQUENCE
CC*    CALL IPLOT (IPN,NUMN,NTBL, IPB,NUMB,NAME,NUMBER)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    IPN,NUMN,NTBL  - DATA BLOCK NAME TABLE (SEE SUBROUTINE NAME1)*
CC*    IPB,NUMB,NAME,NUMBER - PROCESSING BLOCK NAME AND NUMBER TABLE *
CC*                      (SEE SUBROUTINE PROCB1)
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PLOTB
CC*    PLOT
CC*    SYMBOL
CC*    GETREC
CC*    EFPLT
CC*
CC*****

```



SUBROUTINE IPLOT

```

CC***** PLOTX *****
CC*
CC*          SUBROUTINE PLOTX
CC*
CC*    PURPOSE
CC*      TO MOVE THE PEN TO A NEW POSITION AND INTERCEPT PLOT CALLS
CC*      WHICH CAUSE PEN TRAVEL FARTHER THAN TEN FEET. WHEN THIS
CC*      OCCURS, THE PEN MOVEMENT IS BROKEN UP INTO SMALLER MOVES.
CC*      TEN FEET IS THE MAXIMUM TRAVEL ALLOWED IN A SINGLE CALL TO
CC*      THE CALSPAN VERSION OF SUBROUTINE PLOT.
CC*
CC*    CALLING SEQUENCE
CC*      CALL PLOTX (X, Y)
CC*
CC*    DESCRIPTION OF PARAMETERS
CC*      X      - X POSITION TO WHICH PEN IS TO BE MOVED
CC*      Y      - Y POSITION TO WHICH PEN IS TO BE MOVED
CC*
CC*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*      PLOT
CC*
CC*****

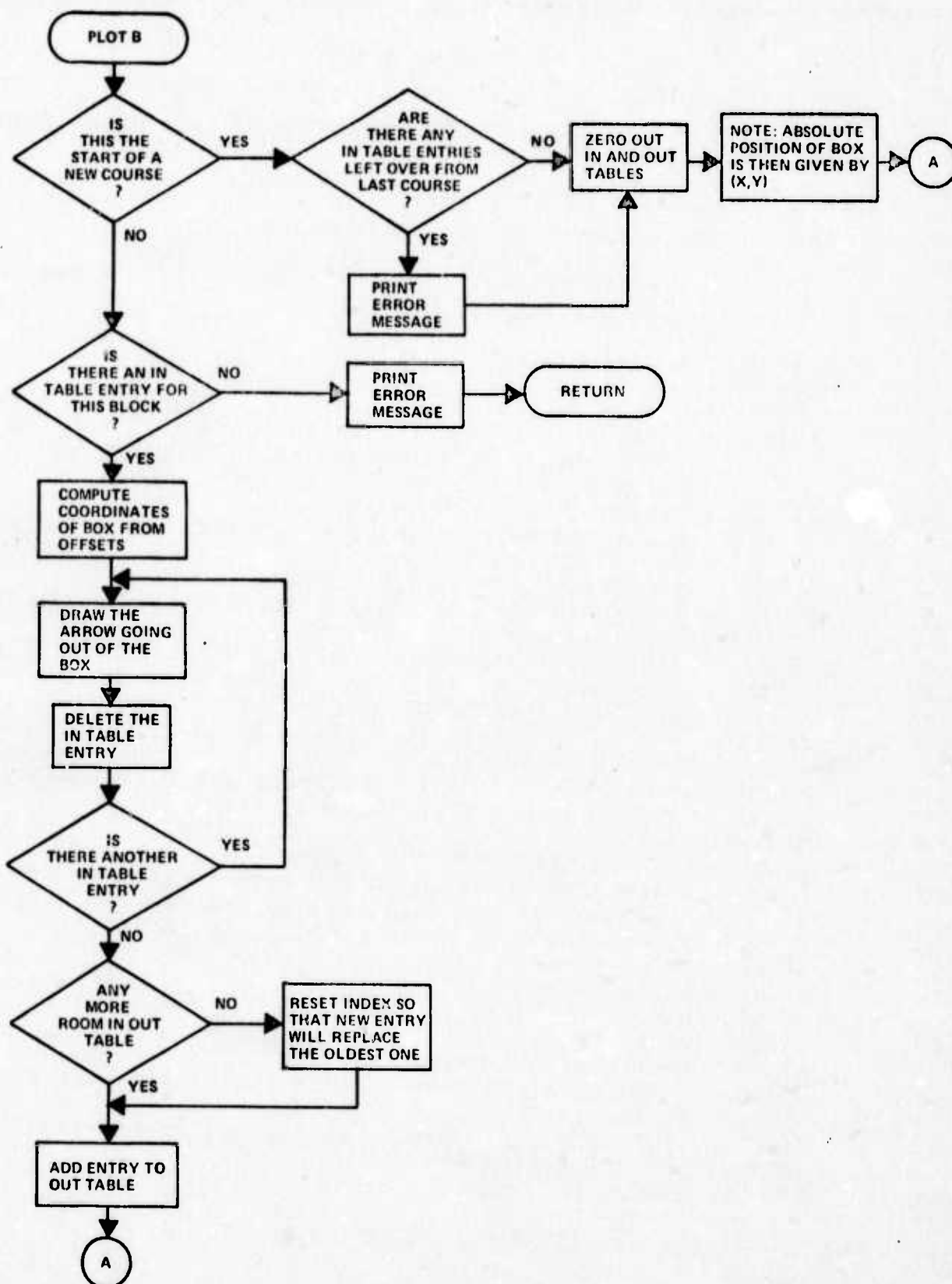
```



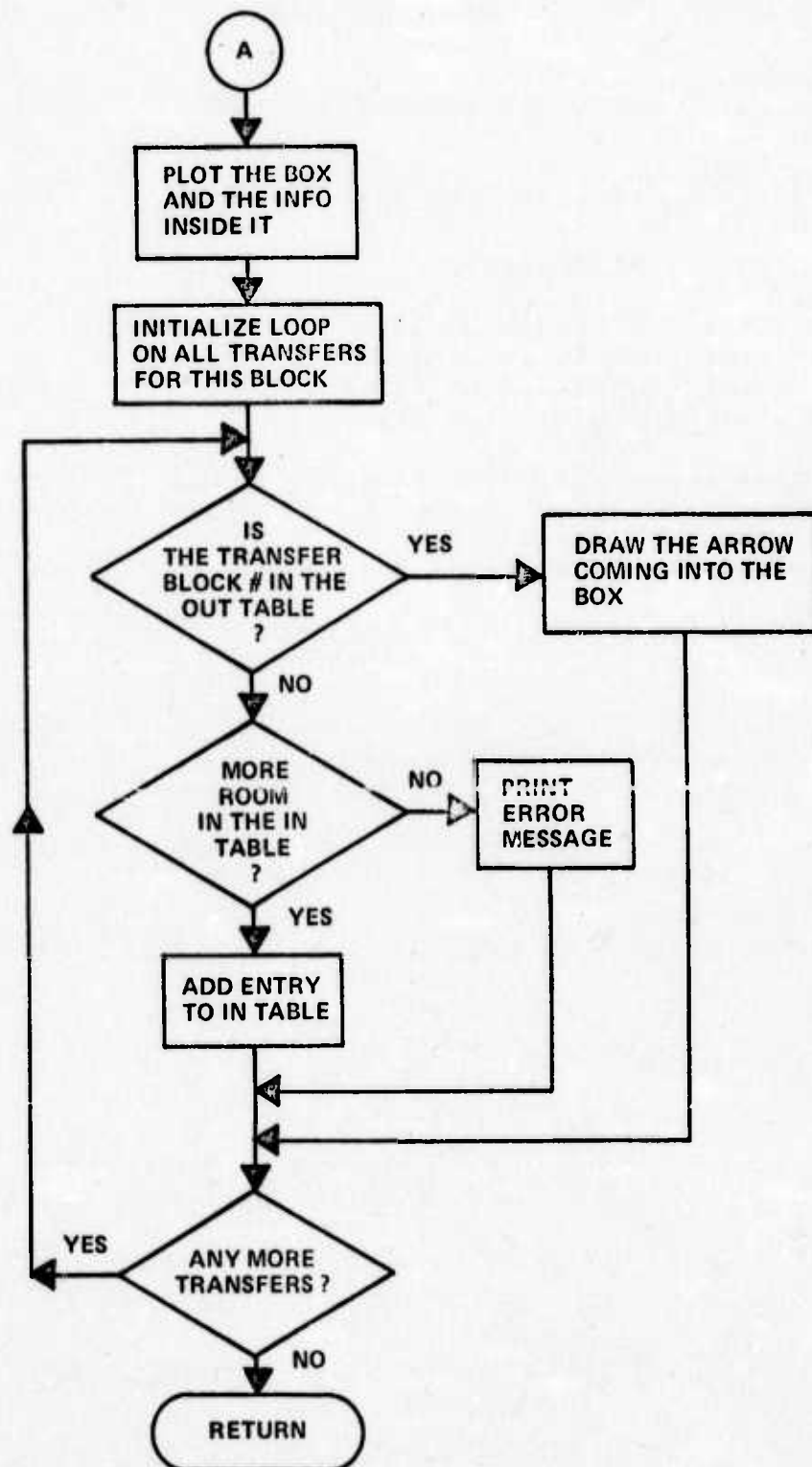
```

CC***** PLOTB *****
CC*
CC*          SUBROUTINE PLOTB
CC*
CC*  PURPOSE
CC*    TO PLOT THE PROCESSING BLOCKS AND CONNECT THEM WITH ARROWS TO
CC*    SHOW THE FLOW
CC*
CC*  CALLING SEQUENCE
CC*    CALL PLOTB (XYIN, IBLKIN, NUMIN, MAXIN,
CC*               XYOUT, IBLKOT, NUMOUT, MAXOUT, ICODE,
CC*               X, Y, IBLKNO, INAME, NTASKS, ITASKS, NTRAN, ITRAN,
CC*               ISYNC, ICSYNC, IBSYNC)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    WORK AREAS
CC*      ***** IN TABLE *****
CC*      XYIN   - ARRAY DIMENSIONED (MAXIN,2), USED TO STORE THE
CC*                COORDINATES OF THE POINTS WHICH NEED ARROWS
CC*                POINTING INTO THEM
CC*      IBLKIN - ARRAY DIMENSIONED (MAXIN), USED TO STORE THE
CC*                BLOCK NUMBER FROM WHICH EACH UNRESOLVED TRANSFER
CC*                IS TO COME FROM
CC*      NUMIN  -- NUMBER OF ENTRIES IN XYIN AND IBLKIN ARRAYS
CC*      MAXIN  - MAXIMUM NUMBER OF ENTRIES IN XYIN AND IBLKIN ARRAYS
CC*      ***** OUT TABLE *****
CC*      XYOUT  - ARRAY DIMENSIONED (MAXOUT,2), USED TO STORE THE
CC*                COORDINATES OF THE START OF THE OUTGOING ARROWS
CC*                FROM EACH PROCESSING BLOCK
CC*      IBLKOT - ARRAY DIMENSIONED (MAXOUT), USED TO STORE BLOCK
CC*                NUMBERS OF ENTRIES IN XYOUT ARRAY
CC*      NUMOUT - NUMBER OF ENTRIES IN THE XYOUT AND IBLKOT ARRAYS
CC*      MAXOUT - MAXIMUM NUMBER OF ENTRIES IN XYOUT ARRAY
CC*  INPUT PARAMETERS
CC*      ICODE - 1 INITIALIZE FOR A NEW COURSE
CC*              2 CONTINUE WORKING ON THE SAME COURSE
CC*      X     - X COORDINATE OF LOWER LEFT CORNER OF BOX
CC*      Y     - Y COORDINATE OF LOWER LEFT CORNER OF BOX
CC*      IBLKNO - BLOCK NUMBER
CC*      INAME  - BLOCK NAME
CC*      NTASKS - NUMBER OF TASKS IN THE BLOCK
CC*      ITASKS - ARRAY OF TASK NAMES
CC*      NTRAN  - NUMBER OF TRANSFERS INTO THE BLOCK
CC*      ITRAN  - ARRAY OF BLOCK NUMBERS FROM WHICH EACH TRANSFER
CC*                TAKES PLACE
CC*      ISYNC  - SYNC CODE (0-NO SYNC, 1-SYNC TO, 2-CORRELATE WITH)
CC*      ICSYNC - SYNC COURSE NAME
CC*      IBSYNC - SYNC BLOCK NUMBER
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PLOT
CC*    SYMBOL
CC*    NUMBER
CC*    ARROW
CC*    LOOKUP
CC*****

```



SUBROUTINE PLOTB



SUBROUTINE PLOTB - CONTINUED

```

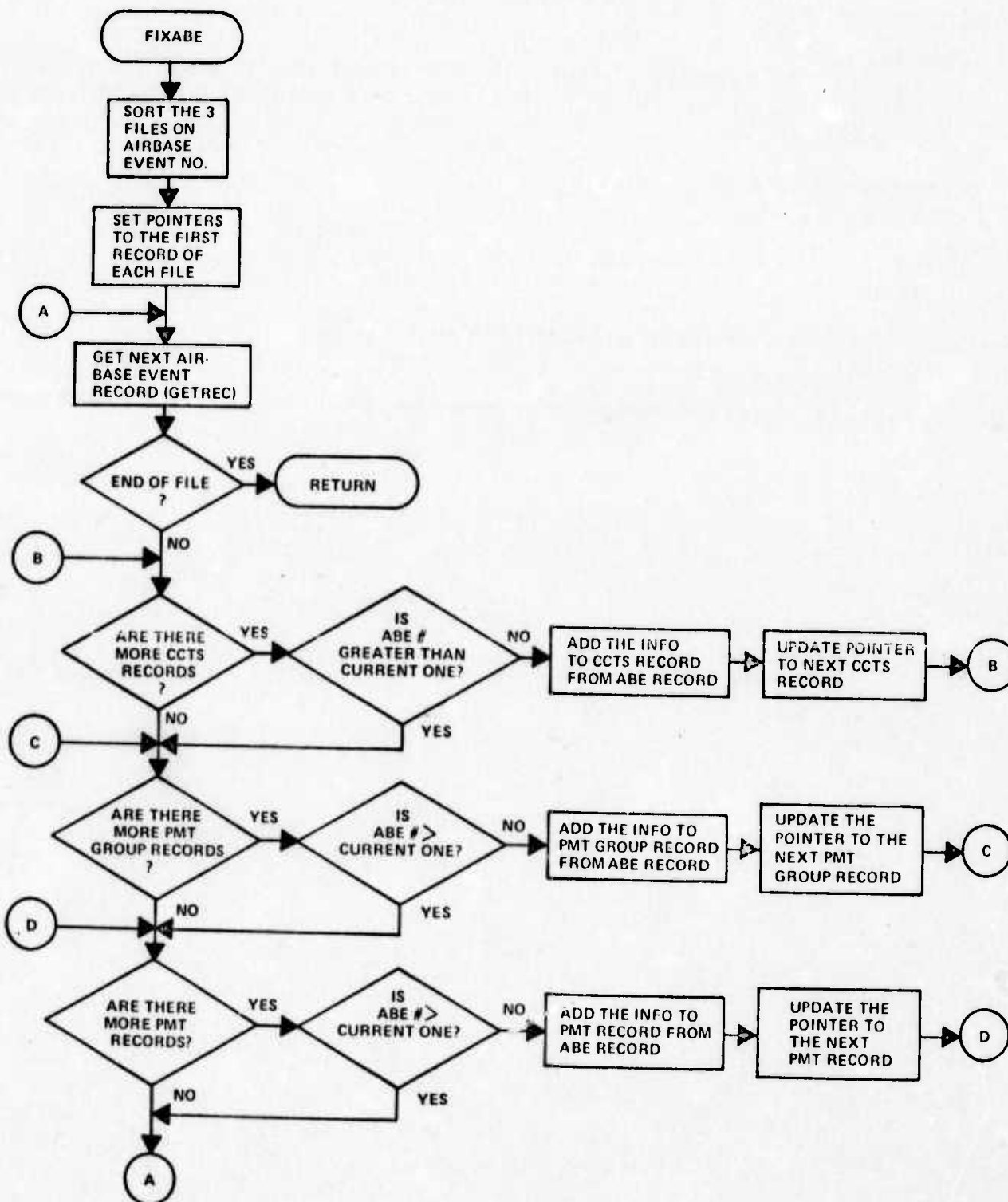
CC***** ARROW *****
CC*
CC*          SUBROUTINE ARROW
CC*
CC*  PURPOSE
CC*    TO DRAW AN ARROW FOR SUBROUTINE PLOTS
CC*
CC*  CALLING SEQUENCE
CC*    CALL ARROW (X1, Y1, X2, Y2)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      X1  - X COORDINATE OF START OF ARROW
CC*      Y1  - Y COORDINATE OF START OF ARROW
CC*      X2  - X COORDINATE OF END POINT
CC*      Y2  - Y COORDINATE OF END POINT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PLOT
CC*
CC*****

```

```

CC***** FIXABE *****
CC*
CC* SUBROUTINE FIXABE
CC*
CC* PURPOSE
CC* TO ADD THE AIRBASE NUMBER AND TIME(FROM THE ASSOCIATED AIRBASE*
CC* EVENT RECORD) TO THE CCTS RECORDS, PMT GROUP RECORDS, AND THE *
CC* PMT RECORDS
CC*
CC* CALLING SEQUENCE
CC* CALL FIXABE
CC*
CC* DESCRIPTION OF PARAMETERS
CC* NONE
CC*
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC* GETREC
CC*
CC*****

```



SUBROUTINE FIXABE



```

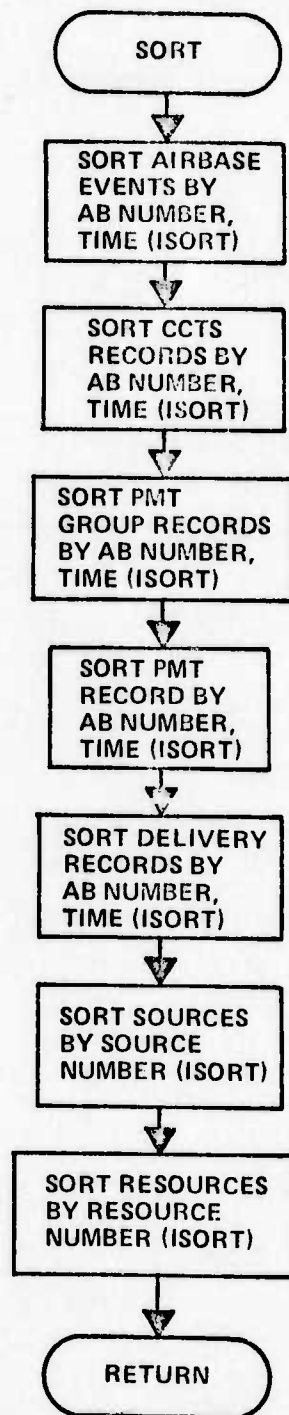
CC***** .FIXPRO *****
CC*
CC*          SUBROUTINE FIXPRO
CC*
CC*  PURPOSE
CC*    TO REPLACE THE SYNC TYPE IN THE PROCESSING BLOCKS WITH THE
CC*    COURSE NUMBER TO WHICH THE BLOCK BELONGS. THIS IS DONE
CC*    FOR USE BY PHASE 3
CC*
CC*  CALLING SEQUENCE
CC*    CALL FIXPRO (IP1, INUM)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    IP1   - ARRAY OF POINTERS TO THE FIRST PROC BLOCK IN EACH
CC*           COURSE
CC*    INUM  - NUMBER OF PROCESSING BLOCKS IN EACH COURSE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*
CC*****

```

```

CC***** SORT *****
CC*
CC*          SUBROUTINE SORT
CC*
CC*  PURPOSE
CC*    TO SORT THE DATA BLOCKS SO THAT THEY ARE IN THE PROPER
CC*    SEQUENCE FOR STEP 2 OF TRAM MODEL
CC*
CC*  CALLING SEQUENCE
CC*    CALL  SORT
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    ISORT
CC*
CC*****

```

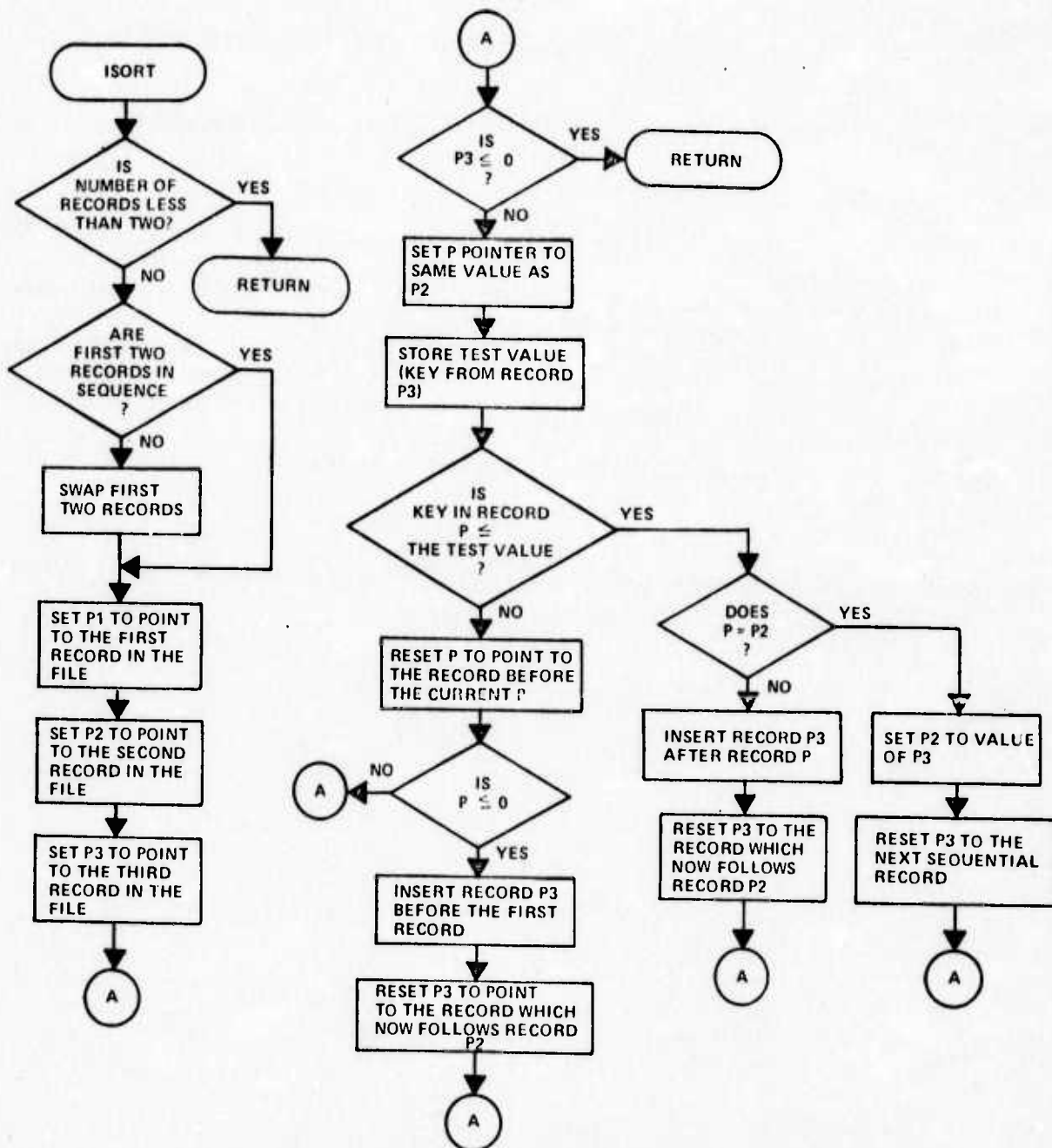


SUBROUTINE SORT

```

CC***** ISORT *****
CC*
CC*          SUBROUTINE ISORT
CC*
CC*  PURPOSE
CC*    TO SORT THE RECORDS OF A GIVEN FILE IN COMMON /FILE/ INTO
CC*    ASCENDING SEQUENCE
CC*
CC*  CALLING SEQUENCE
CC*    CALL ISORT (INDEX, IWORD)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      INDEX  - FILE NUMBER TO BE SORTED
CC*      IWORD  - WORD NUMBER WITHIN THE RECORDS ON WHICH TO SORT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```



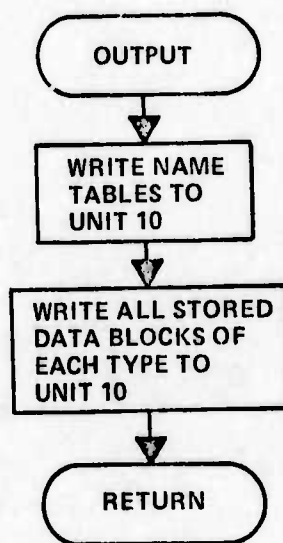
SUBROUTINE ISORT

```

CC***** OUTPUT *****
CC*
CC*          SUBROUTINE OUTPUT
CC*
CC*  PURPOSE
CC*    TO WRITE THE INPUT DATA ONTO THE FILE FOR TRAM STEP 2
CC*
CC*  CALLING SEQUENCE
CC*    CALL  OUTPUT (IPN,NUMN,NTBL, IPB,NUMB,NAME,NUMBER)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IPN,NUMN,NTBL  - DATA BLOCK NAME TABLE (SEE SUBROUTINE NAME1)*
CC*    INPUT-OUTPUT
CC*      IPB,NUMB,NAME,NUMBER - PROCESSING BLOCK NAME AND NUMBER
CC*                          TABLE (SEE SUBROUTINE PROCB1)
CC*      NOTE - PROCESSING BLOCK NAME TABLE IS DESTROYED BY
CC*             THE PROCESS OF CONVERTING PROCESSING BLOCK
CC*             NUMBER TABLE TO CHARACTER FORMAT FOR OUTPUT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    GETREC
CC*    WRITE
CC*
CC*****

```





SUBROUTINE OUTPUT

```

CC***** WRITE *****
CC*
CC*          SUBROUTINE WRITE
CC*
CC*  PURPOSE
CC*    TO WRITE OUT AN ARRAY FOR SUBROUTINE OUTPUT
CC*
CC*  CALLING SEQUENCE
CC*    CALL WRITE (LU, IARRAY, NWDS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      LU      - FORTRAN LOGICAL UNIT NUMBER
CC*      IARRAY - DATA ARRAY
CC*      NWDS    - NUMBER OF WORDS IN IARRAY
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** LOOKUP *****
CC*
CC*          SUBROUTINE LOOKUP
CC*
CC* PURPOSE
CC*   TO LOOK UP A VALUE IN A TABLE AND RETURN ITS POSITION
CC*
CC* CALLING SEQUENCE
CC*   CALL LOOKUP (IVAL, IARRAY, N, ICODE, INDEX)
CC*
CC* DESCRIPTION OF PARAMETERS
CC*   INPUT
CC*     IVAL - VALUE TO BE SEARCHED FOR
CC*     IARRAY- TABLE OF VALUES TO BE SEARCHED FOR
CC*     N    - NUMBER OF ENTRIES IN IARRAY
CC*     ICODE - 1 - DATA VALUES OCCUPY ONE WORD
CC*              2 - DATA VALUES OCCUPY THREE WORDS, USED FOR 10
CC*                 CHARACTER FIELDS ON IBM COMPUTER
CC*                 (REQUIRES IVAL(3), IARRAY(3,N) )
CC*
CC*   OUTPUT
CC*     INDEX - INDEX OF THE VALUE IN THE TABLE, ZERO IF THE VALUE
CC*              IS NOT FOUND
CC*
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*   NONE
CC*
CC*****

```

## Section 1.6

### SUBROUTINE CROSS REFERENCE TABLE

In the table on the following pages, the column headings show the subroutine names that do the calling, and the row headings give the subroutine names that are called.

# SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE1 \*\*\*\*\*

ROUTINE  
OR ENTRY

## USAGE SUMMARY

	MAIN	*BLOCK	TBL	TBL3	TBL4	TBL5	PTYPE	TBL6	TBL7	TBL8	TBL9	TBL10
ADDERC												
ANLOW												
ENDCUR												
FIXABE	X											
FIXPRO	X											
GETREC				X	X	X		X	X	X	X	X
INPUT	X											
IPLOT	X											
ISORT												
ITABLE	X											
LOOKUP												
NAME1	X											
NAME2	X											
OUTPUT	X											
PLOTB												
PLOTX												
PRINC												
PROCB1	X											
PROCB2	X											
PTYPE												
READC												
READTB	X											
REPLC1												
REPLC2												
RPT												
SHIFTR												
SURT	X											

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*

USAGE SUMMARY

ROUTINE OR ENTRY	TBL2	TBL12	TBL13	TBL14	ADDRC	GETREC	TRNSFR	FIXABE	OUTPUT	WRITE	IPLUT	PLUTX
ADDRC												
ARROW												
ENDCOR												
FIXABE												
FIXPRO												
GETREC	X	X	X	X				X	X		X	
INPUT												
IPLUT												
ISORT												
ITABLE								X				
LOOKUP												
NAME1												
NAME2												
OUTPUT												
PLOTB											X	
PLOTX											X	
PRINTC												
PROCB1												
PROCB2												
PTYPE												
READC												
READTB												
REPLC1												
REPLC2												
RPT												
SHIFTR												
SORT												



SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* INDEX \*\*\*\*\*

ROUTINE OR ENTRY

	ADOREC	ARROW	ITABLE	NAME1	NAME2	REPLC1	REPLC2	PROCB1	PROCB2	LOGNUP	SORT	ISORT
ROUTINE OR ENTRY												
ARROW	X											
ENDCOR												
FIXABE												
FIXPRO												
GETREC				X		X	X	X	X			
INPUT												
IPLUT												
ISORT											X	
ITABLE												
LOOKUP	X			X		X	X	X	X			
NAME1												
NAME2												
OUTPUT												
PLOTB												
PLOTX	X											
PRINTC												
PROCB1												
PROCB2												
PTYPE												
READC												
READT9												
REPLC1												
REPLC2												
RPT												
SHIFTR												
SORT												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*  
ROUTINE OR ENTRY USAGE SUMMARY

ROUTINE OR ENTRY	READTB	READC	PRINTC	TEST	STORE	RPT	INPUT	SHIFTR	ENDCOR	FIXPRO
ADDRC										
ARROW										
ENDCOR										
FIXABE										
FIXPRO										
GETREC										
INPUT										
IPLOT										
ISORT										
ITABLE										
LOCKUP										
NAME1										
NAME2										
OUTPUT										
PLOTB										
PLOTX										
PRINC										
PROC81										
PROC82										
PTYPE										
READC										
READTB										
REPLC1										
REPLC2										
RPT										
SHIFTR										
SORT										

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	MAIN	*BLOCK	TBL	TBL3	TBL4	TBL5	PTYPE	TBL6	TBL7	TBL8	TBL9	TBL10
STORE												
TBL	X											
TBL10			X									
TBL12			X									
TBL13			X									
TBL14			X									
TBL2			X									
TBL3			X									
TBL4			X									
TBL5			X									
TBL6			X									
TBL7			X									
TBL8			X									
TEL9			X									
TEST												
TRANSFR												
WRITE												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE1 \*\*\*\*\*  
 ROUTINE OR ENTRY USAGE SUMMARY

	TBL2	TBL12	TBL13	TBL14	ADDREC	GETREC	TRNSFR	FIXABE	OUTPUT	WRITE	IPLOT	PLOTX
STORE												
TBL												
TBL10												
TBL12												
TBL13												
TBL14												
TBL2												
TBL3												
TBL4												
TBL5												
TBL6												
TBL7												
TBL8												
TBL9												
TEST												
TRNSFR												
WRITE												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* PAGE 1 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	PLOTS	ARROW	ITABLE	NAME1	NAME2	REPLC1	REPLC2	PROCB1	PROCB2	LOOKUP	SORT	ISORT
STORE												
TBL												
TBL10												
TBL12												
TBL13												
TBL14												
TBL2												
TBL3												
TBL4												
TBL5												
TBL6												
TBL7												
TBL8												
TBL9												
TEST												
TRANSFR												
WRITE												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	READR	READC	PRINTC	TEST	STORE	RPT	INPUT	SHIFTR	ENDCOR	FIXPRG
STORE							X			
TBL										
TBLIC										
TBL12										
TBL13										
TBL14										
TBL2										
TBL3										
TBL4										
TBL5										
TBL6										
TBL7										
TBL8										
TBL9										
TEST						X	X			
TRNSFR							X			
WRITE										



## Section 1.7

### COMMON VARIABLE DEFINITIONS

The tables on the following pages define the meaning of each variable contained in each of the common blocks used by this program.

```

*****
*
*   COMMON /FILE/   -   INTERNAL STORAGE FOR DATA BLOCKS
*
*****
*
*   VARIABLE      *   DESCRIPTION
*
*****
*   MAXFLE      *   MAXIMUM NUMBER OF BLOCK TYPES WHICH CAN BE STORED
*   ISTRT(J)    *   POINTER TO THE FIRST BLOCK STORED OF EACH TYPE
*   IEND(J)     *   POINTER TO THE LAST BLOCK STORED OF EACH TYPE
*   NRECS(J)    *   NUMBER OF BLOCKS CURRENTLY STORED FOR EACH TYPE
*   MAXADR      *   DIMENSION OF VARIABLE IFILE
*   NEXTAD      *   POINTER TO THE NEXT AVAILABLE LOCATION IN THE IFILE ARRAY
*   IFILE(I)    *   STORAGE AREA FOR THE DATA BLOCKS.  EACH BLOCK HAS THE
*                 *   FOLLOWING FORMAT
*                 *   WORD NUMBER      DESCRIPTION
*                 *       1           POINTER TO THE NEXT BLOCK
*                 *       2           POINTER TO THE LAST BLOCK
*                 *       3           NUMBER OF WORDS IN THIS BLOCK
*                 *       4           DATA WORDS
*                 *       .
*                 *       .
*                 *       .
*                 *   3+NWDS
*
*****

```

```

*****
*
*   COMMON /TABLES/ - FUNCTION NAME LOOKUP TABLES
*
*****
*
*   VARIABLE      *   DESCRIPTION
*
*****
*   NTBLS          *   NUMBER OF TABLES
*   ITBLS(3,I,J)  *   FUNCTION NAME LOOKUP TABLES
*                   *   SUBSCRIPT      DESCRIPTION
*                   *       1          ALLOWS 3 WORDS PER TEN CHARACTER NAME
*                   *       2          INDEX OF EACH ENTRY IN A TABLE
*                   *                   NOTE - THE FIRST ENTRY (ITBLS(1,1,J)) GIVES THE
*                   *                   NUMBER OF NAMES, AND THE REST OF THE ENTRIES
*                   *                   CONTAIN THE NAMES
*                   *       3          INDEX OF TABLE NUMBER
*
*****

```

```

*****
*
*   COMMON  /ICNAME/  -   CARD NAME TABLE
*
*****
*
*   VARIABLE      *   DESCRIPTION
*
*****
*   NNAMES        *   NUMBER OF CARD TYPES
*   INAMES(3,J)   *   CARD NAME OF EACH CARD TYPE
*                 *   (3 WORDS PER 10 CHARACTER NAME, BLANK FOR THOSE CARDS WHICH
*                 *   ARE IDENTIFIED BY A HEADER CARD)
*   IBLKNO(J)     *   INTERNAL DATA BLOCK NUMBER ASSOCIATED WITH EACH INPUT CARD
*
*****

```

```

*****
*
*   COMMON /DDIN/   -   INPUT DATA CARD DESCRIPTION TABLE
*
*****
*           *
*  VARIABLE *  DESCRIPTION
*           *
*****
*           *
*N1(J)      *  NUMBER OF CHARACTER DATA FIELDS CONTAINED ON EACH CARD
*           *  TYPE (INCLUDES CARD NAME FIELD)
*N2(J)      *  NUMBER OF NUMERIC PARAMETERS CONTAINED ON EACH CARD
*IRNG1(I,J)*  LOWER BOUND OF ACCEPTANCE RANGE FOR EACH NUMERIC VALUE
*           *  ON EACH CARD TYPE
*IRNG2(I,J)*  UPPER BOUND OF ACCEPTANCE RANGE FOR EACH NUMERIC VALUE
*           *  ON EACH CARD TYPE
*           *
*****

```



```

*****
*
*   COMMON /DBD/   -   INTERNAL DATA BLOCK DESCRIPTION TABLE
*
*****
*
*   VARIABLE *   DESCRIPTION
*
*****
*
*   NELKS      *   NUMBER OF DATA BLOCKS
*   ID1(I)     *   POINTER TO START OF DATA BLOCK DESCRIPTION (IN IFMT ARRAY)
*               *   FOR EACH DATA BLOCK
*   ID2(I)     *   NUMBER OF WORDS OF DESCRIPTION FOR EACH DATA BLOCK
*   IFMT(J)    *   DATA BLOCK DESCRIPTION CODES
*               *   THE FOLLOWING CODES ARE CURRENTLY BEING USED
*
*   CODE      MEANING
*   -N        DATA WORD CONTAINS THE NUMBER OF ENTRIES OF VARIABLE
*               LENGTH DATA TO FOLLOW.  N GIVES THE NUMBER OF DATA
*               WORDS PER ENTRY, AND THE NEXT N CODES DESCRIBE THE
*               ENTRIES
*   0         NO DATA, OR A CONTINUATION
*   1         INTEGER
*   4         FLOATING POINT
*   8         CHARACTER DATA
*   9         DATA BLOCK NAME (MUST BE UNIQUE)
*   10        DATA BLOCK NAME (MAY NOT BE UNIQUE)
*   *11-30    BLOCK NAME REFERENCE
*               CODE MINUS 10 GIVES THE BLOCK NUMBER
*   *31-50    BLOCK NAME REFERENCE (MAY BE BLANK)
*               CODE MINUS 30 GIVES THE BLOCK NUMBER
*   *51-60    FUNCTION NAME
*               CODE MINUS 50 GIVES TABLE NUMBER
*   *61-70    FUNCTION NAME (MAY BE BLANK)
*               CODE MINUS 60 GIVES TABLE NUMBER
*
*****

```



## Section 1.8

### INTERNAL DATA BLOCK DESCRIPTIONS

The tables on the following pages define the contents of each of the data blocks used to store the TRAM inputs in phase 1. These data blocks are stored in common area /FILE/. The format code associated with each data word is used by the program to determine what data are contained in that word. See the description of common block /DBD/ for the definition of these codes.

```

*****
*
*   DATA BLOCK NUMBER 1 - CONTROL PARAMETERS (NOT CURRENTLY USED)
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 * 4 * ATTENTION RATIO
* 5 * 4 *
* 6 * 4 *
* 7 * 4 *
* 8 * 1 * DELAY TIME CONSTANT
* 9 * 1 *
* 10 * 1 *
* 11 * 1 *
* 12 * 4 * PERCENT COPILOTS RECOVERABLE AS PILOTS
* 13 * 1 * COPILOT HOLDING PERIOD
* 14 * 1 * NUMBER OF CALENDAR UNITS/YEAR
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 2 - AIR BASES
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 * 9 * AIR BASE NAME
* 5 * 0 *
* 6 * 0 *
* 7 * 1 * INITIAL INVENTORY OF AIRCRAFT
* 8 * 1 * INITIAL NUMBER OF PILOTS
* 9 * 1 * INITIAL NUMBER OF COPILOTS
* 10 * 1 * INITIAL NUMBER OF OSO
* 11 * 1 * INITIAL NUMBER OF DSO
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 3 - RESOURCE INVENTORY
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 * 10 * RESOURCE NAME
* 5 * 0  *
* 6 * 0  *
* 7 * 51 * GENERATING FUNCTION NAME
* 8 * 0  *
* 9 * 0  *
* 10 * 1 * START DATE
* 11 * 1 * END DATE
* 12 * -1 * NUMBER OF PARAMETERS
* - * 1 * PARAMETERS
*   *   *
*****

```



```

*****
*
*   DATA BLOCK NUMBER 4 - SOURCE
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *   *
*   1 *   * * CARD SEQUENCE NUMBER
*   2 *   * * NOT USED
*   3 *   * * NOT USED
*   4 * 10 * * NAME OF SOURCE
*   5 * 0  *
*   6 * 0  *
*   7 * 51 * * GENERATING FUNCTION NAME
*   8 * 0  *
*   9 * 0  *
*  10 * 1  * * START DATE
*  11 * 1  * * END DATE
*  12 * -1 * * NUMBER OF PARAMETERS
*   - * 1  * * PARAMETERS
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 5 - AIRCRAFT DELIVERIES
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   * CODE *
*   *   *
*****
*   *   *
* 1 *   * * CARD SEQUENCE NUMBER
* 2 *   * * NOT USED
* 3 *   * * NOT USED
* 4 *   * 12 * AIR BASE NAME
* 5 *   * 0 *
* 6 *   * 0 *
* 7 *   * 1 * DATL
* 8 *   * 1 * NUMBER OF A/C
*   *   *
*****

```



```

*****
*
*   DATA BLOCK NUMBER 6 - COURSE BLOCK
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 * 9 * COURSE NAME
* 5 * 0 *
* 6 * 0 *
* 7 * 1 * COURSE TYPE
* 8 * 1 * PERSONNEL TYPE
* 9 * 4 * X
* 10 * 4 * Y
* 11 * 1 * MAX CLASS SIZE
* 12 * 1 * CLASS PERIOD
* 13 * 1 * PRIORITY
* 14 * 1 * EARLIEST GRADUATION DATE
* 15 * 1 * POINTER TO FIRST PROC BLOCK IN THIS COURSE
* 16 * 1 * POINTER TO LAST PROC BLOCK IN THIS COURSE
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 7 - PROC BLOCK
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *   *
* 1 *   * * CARD SEQUENCE NUMBER
* 2 *   * * NOT USED
* 3 *   * * NOT USED
* 4 *   8 * * PROC BLOCK NAME
* 5 *   0 *
* 6 *   0 *
* 7 * 36 * * SYNC COURSE NAME
* 8 *   0 *
* 9 *   0 *
* 10 * 1 * * SYNC BLOCK NO
* 11 * 1 * * SYNC CODE
* 12 * 1 * * BLOCK NUMBER
* 13 * 1 * * DURATION
* 14 * 4 * * X
* 15 * 4 * * Y
* 16 * 1 * * PRIORITY
* 17 * -3 * * NUMBER TRANSFERS
* 18 * -3 * * NUMBER TASKS
* - * 1 * * BLOCK NUMBER TRANSFERRED FROM
* - * 1 * * PRIORITY
* - * 4 * * RATIO
* - * 18 * * TASK NAME
* - * 0 *
* - * 0 *
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 8 - TASK
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 *   9 * TASK NAME
* 5 *   0 *
* 6 *   0 *
* 7 * 52 * TASK FUNCTION NAME
* 8 *   0 *
* 9 *   0 *
* 10 * 39 * RUB NAME
* 11 *   0 *
* 12 *   0 *
* 13 *   1 * TASK TYPE
* 14 * -1 * NUMBER OF PARAMETERS
* - *   1 * PARAMETERS
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER  9 -  RUB
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *   *
*   *   *   *
*****
*   *   *
*   1 *   *   * CARD SEQUENCE NUMBER
*   2 *   *   * NOT USED
*   3 *   *   * NOT USED
*   4 *   *   9   * RUB NAME
*   5 *   *   0   *
*   6 *   *   0   *
*   7 *   *  -3   * NUMBER OF RESOURCES
*  - *   *  20   * NAME OF RUBB
*  - *   *   0   *
*  - *   *   0   *
*   *   *   *
*****

```



```

*****
*
*   DATA BLOCK NUMBER 10 -  RUDB
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   * CODE *
*   *   *
*****
*   *   *
* 1 *   * * CARD SEQUENCE NUMBER
* 2 *   * * NOT USED
* 3 *   * * NOT USED
* 4 *   * 9 * RUDB NAME
* 5 *   * 0 *
* 6 *   * 0 *
* 7 *   * 33 * RESOURCE NAME
* 8 *   * 0 *
* 9 *   * 0 *
* 10 *   * 63 * GROUPING FUNCTION
* 11 *   * 0 *
* 12 *   * 0 *
* 13 *   * 64 * TIMING FUNCTION
* 14 *   * 0 *
* 15 *   * 0 *
* 16 *   * 39 * SECONDARY RUB
* 17 *   * 0 *
* 18 *   * 0 *
* 19 *   * 40 * ALTERNATE RUDB
* 20 *   * 0 *
* 21 *   * 0 *
* 22 *   * 1 * UNITS CONSUMPTION/UNIT USER
* 23 *   * 34 * SOURCE NAME
* 24 *   * 0 *
* 25 *   * 0 *
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 11 - AIRBASE TIME HISTORY (NOT CURRENTLY USED)
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
* 1 *   * CARD SEQUENCE NUMBER
* 2 *   * NOT USED
* 3 *   * NOT USED
* 4 * 12 * AIRBASE NAME
* 5 * 0 *
* 6 * 0 *
* 7 * 1 * PERSONNEL TYPE
* 8 * -1 * NUMBER OF POINTS
* - * 1 * POINTS
*   *   *
*****

```



```

*****
*
*   DATA BLOCK NUMBER 12 - AIRBASE EVENT
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
* 1 *   * * CARD SEQUENCE NUMBER
* 2 *   * * NOT USED
* 3 *   * * NOT USED
* 4 *   * 9 * EVENT NAME
* 5 *   * 0 *
* 6 *   * 0 *
* 7 *   * 12 * AIRBASE NAME
* 8 *   * 0 *
* 9 *   * 0 *
* 10 *   * 1 * TIME
* 11 *   * 4 * CREW RATIO
* 12 *   * 4 * ALERT RATIO
* 13 *   * 1 * HRS/CREW/WEEK
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 13 - CCTS BLOCK
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *   *
* 1 *   * * CARD SEQUENCE NUMBER
* 2 *   * * NOT USED
* 3 *   * * NOT USED
* 4 * 22 * * AIRBASE EVENT NAME
* 5 * 0  * *
* 6 * 0  * *
* 7 * 16 * * COURSE NAME
* 8 * 0  * *
* 9 * 0  * *
* 10 * 1  * * PERSONNEL TYPE
* 11 * 4  * * RATIO
* 12 * 0  * * A B NUMBER
* 13 * 0  * * TIME
*   *   *
*****

```

```

*****
*
*   DATA BLOCK NUMBER 14 - PMT GROUP
*
*****
*   *
* WORD * FORMAT * DESCRIPTION
*   *   CODE   *
*   *   *
*****
*   *
*   1 *          * CARD SEQUENCE NUMBER
*   2 *          * NOT USED
*   3 *          * NOT USED
*   4 *          * AIRBASE EVENT NAME
*   5 *          *
*   6 *          *
*   7 *          * PERIOD
*   8 *          * NUMBER OF PMT COURSES
*   9 *          * AB NUMBER
*  10 *          * TIME
*  11 *          * PMT NUMBER
*
*****

```

```

*****
*
*   DATA BLOCK NUMBER 15 - PMT RECORD
*
*****
*   *   *
* WORD * FORMAT * DESCRIPTION
*   *   *
*   *   *
*****
*   *   *
*   1 *   * * CARD SEQUENCE NUMBER
*   2 *   * * NOT USED
*   3 *   * * NOT USED
*   4 *   22 * AIRBASE EVENT NAME - COPIED FROM PMT GROUP CARD
*   5 *   0 *
*   6 *   0 *
*   7 *   16 * COURSE NAME
*   8 *   0 *
*   9 *   0 *
*  10 *   1 * PERSONNEL TYPE
*  11 *   4 * RATIO
*  12 *   1 * TIME LOST
*  13 *   0 * A B NUMBER
*  14 *   0 * TIME
*  15 *   0 * PMT NUMBER
*   *   *
*****

```



## Section 1.9

### COMMON VARIABLE CROSS REFERENCE TABLE

The table on the following pages shows how each subroutine uses each common variable. The subroutine names are printed across the top of the table, and the variable names down the left side.

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE I \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	*BLOCK	TBL	TBL3	TBL4	TBL5	PTYPE	TBL6	TBL7	TBL8	TBL9	TBL10
ISLKN0	I											
ID1	I											
ID2	I											
IEND	I											
IFILE	I											
IFMT	I											
INAMES	I											
IPLOT	I											
IRNG1	I											
IRNG2	I											
ISTRT	I											
ITBLS	I											
MAXRDR	I											
MAXFLE	I											
NBLKS	I											
NEXTAD	I											
NNAMES	I											
NRECS	I											
NTBLS	I											
N1	I											
N2	I											



CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE I \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	TBL2	TBL12	TEL13	TBL14	ADDRC	GETREC	TRNSFR	FIXAGE	OUTPUT	WRITE	IPLUT	PLCTX
IBLKND	I										A S	
ID1	I											
ID2	I											
IEND	I	C	C	C	FSC	C		C	C		C	
IFILE	I	F CE	F CE	F CE	SCE	F CE		FSCE	FSCE		A F CE	
IFMT	I											
INAMES	I											
IPLOT	I											
IRNG1	I											
IRNG2	I											
ISTR1	I	C	C	F C	SC	F C		F C	C		F C	
ITBLS	I								A F C			
MAXADR	I	C	C	C	F C	C		C	C		F C	
MAXFLE	I	C	C	C	F C	C		C	C		C	
NSLKS	I											
NEXTAD	I	C	C	C	FSC	C		C	C		F C	
MNAMES	I											
NRECS	I	C	C	C	FSC	C		C	F C		C	
NTBLS	I											
N1	I										FS	
N2	I										FS	

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	PLUTB	ARROW	ITABLE	NAME1	NAME2	REPLC1	REPLC2	PROCB1	PROCB2	LOOKUP	SCRT	ISORT
IBLKNO	A F											
ID1			F C	F C	F C							
ID2			F C	F C	F C							
IEND				C		C		C				C
IFILE				FSCE		FSCE		FSCE	A FSCE			FSCE
IFMT			F C	F C	F C							
INAMES												
IPLOT												
IRNG1												
IRNG2												
ISTR1				C	A S	C		C				FSC
ITBLS			A C									
MAXADR				C		C		C				C
MAXELE				C		C		C				C
NELKS			F C	F C	F C							C
NEXTAD				C		C		C				
NNAMES												
NRECS				C		C		F C				F C
RTBLS			C									
N1										FS		
N2												

CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE I \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	READTB	READC	PRINC	TEST	STORE	RPT	INPUT	SHIFTR	ENDCOR	FIXPRO
IBLKND	I	SC			C			FC			
ID1	I	SC									
ID2	I	SC									
LEND	I					C		C		FC	C
IFILE	I					A	SCE	A	SCE	FSCE	SCE
IFMT	I	SC									
INAMES	I	SC			FC			A	C		
IPLOT	I							A	S		
IRNG1	I	SC	C	C	FC			C			
IRAG2	I	SC	C	C	FC			C			
ISTR1	I					C		C		C	C
ITBLS	I	SC									
MAXADR	I					C		C		C	C
MAXFLE	I					C		C		C	C
NBLKS	I	FSC									
NEXTAD	I					C		C		C	C
NNAMES	I	FSC			C			A	C		
NRECS	I					C		FC		FC	C
NTBLS	I	FSC									
N1	I	SC	FC	FC	FC			FC	A	F	
N2	I	SC	FC	FC	FC			FC	A	F	

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE1 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

SYMBOL	TYPE	READTB	READC	PRINTC	TEST	STORE	RPT	INPUT	SHIFTR	ENDCOR	FIXPRO
IBLKND	I	SC			C			F C			
ID1	I	SC									
ID2	I	SC									
IEND	I					C		C		F C	C
IFILE	I					A SCE		A SCE		FSCE	SCE
IFMT	I	SC									
INAMES	I	SC			F C			A C			
IPLOT	I							A S			
IRMG1	I	SC	C	C	F C			C			
IRMG2	I	SC	C	C	F C			C			
ISTRT	I					C		C		C	C
ITBLS	I	SC									
MAXADR	I					C		C		C	C
MAXFLE	I					C		C		C	C
NBLKS	I	FSC									
NEXTAD	I					C		C		C	C
NNAMES	I	FSC			C			A C			
NRECS	I					C		F C		F C	C
NTBLS	I	FSC									
N1	I	SC	F C	F C	F C			F C			A F
N2	I	SC	F C	F C	F C			F C			A F



# CROSS REFERENCE USAGE CODES

## A ARGUMENT

THE SYMBOL IS A VARIABLE OR FUNCTION NAME WHICH APPEARS IN AN ARGUMENT LIST OF A CALL, SUBROUTINE, FUNCTION, OR ENTRY STATEMENT.

## D DATA INITIALIZATION

THE SYMBOL IS A VARIABLE WHICH IS INITIALIZED IN A DATA OR TYPE SPECIFICATION STATEMENT SUCH AS A COMPLEX SPECIFICATION STATEMENT.

## F FEELCH A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE MOST RECENTLY ASSIGNED VALUE IS ACCESSED BUT NOT CHANGED.
2. FUNCTION NAME OR ARGUMENT OF A FUNCTION WHICH APPEARS ON THE RIGHT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT OR APPEARS IN AN IF STATEMENT TEST.
3. DUMMY ARGUMENT IN A STATEMENT FUNCTION DEFINITION.

## S STORE A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE VALUE IS REPLACED BY ANOTHER VALUE.
2. FUNCTION NAME WHICH APPEARS ON THE LEFT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT.
3. NAME OF A STATEMENT FUNCTION IN THE DEFINITION OF THAT FUNCTION.

## C COMMON

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A COMMON STATEMENT OR IS THE NAME OF A LABELED COMMON BLOCK.

## E EQUIVALENCE

THE SYMBOL IS A VARIABLE WHICH APPEARS IN AN EQUIVALENCE STATEMENT.

## T TYPE SPECIFICATION

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A:

1. TYPE SPECIFICATION STATEMENT AND IS NOT INITIALIZED IN THAT STATEMENT.
2. DIMENSION OR EXTERNAL STATEMENT.

## N ENTRY POINT

THE SYMBOL IS AN ENTRY POINT DEFINED BY AN ENTRY STATEMENT IN A SUBROUTINE OR FUNCTION.

## X EXTERNAL REFERENCE

THE SYMBOL IS A SUBROUTINE OR ENTRY NAME WHICH APPEARS IN A CALL STATEMENT.

## Section 1.10

### INITIALIZATION FILE

This file is read by subroutine READTB from FORTRAN logical unit 9. It is a formatted file that contains card images, and is used to initialize the following common blocks: ICNAME, TABLES, DDIN, and DBD. Normally, any changes to the values on this file would be accompanied with program modifications.

Four tables are contained on this file, one to initialize each common block. The contents of each of these tables are summarized below. For a detailed description of the values on this file, refer to the descriptions of the common blocks that they initialize. A listing of this file will be provided with the program listings.

TABLE 1 (Initializes common ICNAME)

This table contains the card names, which are used to identify the input cards, and the internal block numbers for the data contained on those cards.

TABLE 2 (Initializes common TABLES)

This table contains the function names to be coded on the input cards.

TABLE 3 (Initializes common DDIN)

This table contains a description of each input card. This includes the number of character fields on the card, the number of numeric fields, and the range of acceptable values for each numeric field.

TABLE 4 (Initializes common DBD)

This table contains a description of the internal data blocks used to store the inputs. This description is used by the program to locate and replace character name references with the proper integer code.



Section 1.11  
OUTPUT FILE DESCRIPTION

The following tables show the contents of the output file from TRAM phase 1. This is an unformatted file that is written onto FORTRAN logical unit 10 for passage to phase 2. The first table summarizes the records that are contained on the file, and their order. Other tables follow, which give a detailed description of those records that contain more than one item.

\*\*\*\*\*

\*  
\* UNIT TEN FILE DESCRIPTION \*  
\*

\*\*\*\*\*

\*  
\* THIS UNFORMATTED (BINARY) FILE CONTAINS THE FOLLOWING RECORDS \*  
\*

- \* - NUMBER OF AIR BASES
- \* - AIR BASE NAMES
- \* - NUMBER OF COURSES
- \* - COURSE NAMES
- \* - NUMBER OF GENERATING FUNCTIONS
- \* - GENERATING FUNCTION NAMES
- \* - NUMBER OF PROCESSING BLOCKS
- \* - PROCESSING BLOCK NAMES
- \* - NUMBER OF PROCESSING BLOCKS
- \* - PROCESSING BLOCK NUMBERS
- \* - NUMBER OF RESOURCES
- \* - RESOURCE NAMES
- \* - NUMBER OF RESOURCE UTILIZATION BLOCKS
- \* - RESOURCE UTILIZATION BLOCK NAMES
- \* - NUMBER OF RESOURCE UTILIZATION DESCRIPTION BLOCKS
- \* - RESOURCE UTILIZATION DESCRIPTION BLOCK NAMES
- \* - NUMBER OF RESOURCE UTILIZATION GROUPING FUNCTIONS
- \* - RESOURCE UTILIZATION GROUPING FUNCTION NAMES
- \* - NUMBER OF RESOURCE UTILIZATION TIMING FUNCTIONS
- \* - RESOURCE UTILIZATION TIMING FUNCTION NAMES
- \* - NUMBER OF SOURCES
- \* - SOURCE NAMES
- \* - NUMBER OF TASK BLOCKS
- \* - TASK BLOCK NAMES
- \* - NUMBER OF TASK FUNCTIONS
- \* - TASK FUNCTION NAMES

\*  
\* CONTINUED ON NEXT PAGE \*  
\*

\*\*\*\*\*

```

*****
*
*   UNIT TEN FILE DESCRIPTION - CONTINUED
*
*****
* - NUMBER OF COURSE RECORDS
* - COURSE RECORDS
* - NUMBER OF PROCESSING BLOCK RECORDS
* - PROCESSING BLOCK RECORDS
* - NUMBER OF TASK BLOCK RECORDS
* - TASK BLOCK RECORDS
* - NUMBER OF RESOURCE UTILIZATION BLOCK RECORDS
* - RESOURCE UTILIZATION BLOCK RECORDS
* - NUMBER OF RESOURCE UTILIZATION DESCRIPTION BLOCK RECORDS
* - RESOURCE UTILIZATION DESCRIPTION BLOCK RECORDS
* - NUMBER OF AIR BASE INVENTORY RECORDS
* - AIR BASE INVENTORY RECORDS (SORTED BY AIR BASE NUMBER)
* - NUMBER OF TIME HISTORY RECORDS
* - TIME HISTORY RECORDS (SORTED BY AIR BASE NUMBER)
* - NUMBER OF AIR BASE EVENT RECORDS
* - AIR BASE EVENT RECORDS (SORTED BY AIR BASE NUMBER AND TIME)
* - NUMBER OF CCTS RECORDS
* - CCTS RECORDS (SORTED BY AIR BASE NUMBER AND TIME)
* - NUMBER OF PMT GROUP RECORDS
* - PMT GROUP RECORDS (SORTED BY AIR BASE NUMBER AND TIME)
* - NUMBER OF PMT COURSE RECORDS
* - PMT COURSE RECORDS (SORTED BY AIR BASE NUMBER, TIME, AND
  PMT GROUP NUMBER)
* - NUMBER OF AIR CRAFT DELIVERY RECORDS
* - AIR CRAFT DELIVERY RECORDS (SORTED BY AIR BASE NUMBER AND TIME)
* - NUMBER OF SOURCE RECORDS
* - SOURCE RECORDS (SORTED BY SOURCE NUMBER)
* - NUMBER OF RESOURCE RECORDS
* - RESOURCE RECORDS (SORTED BY RESOURCE NUMBER)
*
*****

```

```

*****
*
*           C O U R S E   R E C O R D S
*
*****
*
* WORD *   DESCRIPTION
*
*****
*  1 * COURSE TYPE
*  2 * PERSONNEL TYPE
*  3 * PRIORITY
*  4 * MAXIMUM CLASS SIZE
*  5 * CLASS PERIOD
*  6 * EARLIEST GRADUATION DATE
*
*****

```



```

*****
*
*  P R O C E S S I N G   B L O C K   R E C O R D S
*
*****
*      *
* WORD *  DESCRIPTION
*      *
*****
*      *
*  1 *  DURATION
*  2 *  PRIORITY
*  3 *  COURSE NUMBER TO WHICH THIS BLOCK BELONGS
*  4 *  SYNCHRONIZE-CORRELATE BLOCK NUMBER
*  5 *  NUMBER OF TRANSFERS (UP TO 5)
*  6 *      PROCESSING BLOCK NUMBER
*  7 *      PRIORITY
*  8 *      RATIO
*  . *      .
*  . *      .
*  . *      .
* 21 *  NUMBER OF TASKS (UP TO 5)
* 22 *      TASK BLOCK NUMBER
*  . *      .
*  . *      .
*  . *      .
* 26 *      TASK BLOCK NUMBER
*      *
*****

```



```

*****
*
*           TASK BLOCK RECORDS
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * TASK FUNCTION NUMBER
*  2 * TASK TYPE
*  3 * RESOURCE UTILIZATION BLOCK NUMBER
*  4 * NUMBER OF PARAMETERS (UP TO 5)
*  5 *     PARAMETER 1
*  . *     .
*  . *     .
*  . *     .
*  9 *     PARAMETER 5
*      *
*****

```

```

*****
*
*          RESOURCE UTILIZATION BLOCK RECORDS
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * NUMBER OF RESOURCE UTILIZATION DESCRIPTION BLOCKS *
*      * (UP TO 6)
*  2 * RESOURCE UTILIZATION DESCRIPTION BLOCK NUMBER *
*      * .
*      * .
*      * .
*  7 * RESOURCE UTILIZATION DESCRIPTION BLOCK NUMBER *
*      *
*****

```

```

*****
*
*          RESOURCE UTILIZATION DESCRIPTION BLOCK RECORDS
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * RESOURCE NUMBER
*  2 * RESOURCE UTILIZATION GROUPING FUNCTION NUMBER
*  3 * RESOURCE UTILIZATION TIMING FUNCTION NUMBER
*  4 * SECONDARY RESOURCE UTILIZATION BLOCK NUMBER
*  5 * ALTERNATE RESOURCE UTILIZATION DESCRIPTION BLOCK NUMBER
*  6 * UNITS OF CONSUMPTION PER UNIT USER
*      *
*****

```

```

*****
*
*   A I R   B A S E   I N V E N T O R Y   R E C O R D S
*
*****
*
*   WORD *   DESCRIPTION
*
*****
*
*   1 *   INITIAL INVENTORY OF AIR CRAFT
*   2 *   INITIAL INVENTORY OF PILOTS
*   3 *   INITIAL INVENTORY OF COPILOTS
*   4 *   INITIAL INVENTORY OF CSO
*   5 *   INITIAL INVENTORY OF DSO
*
*****

```



```

*****
*
*      T I M E   H I S T O R Y   R E C O R D S
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * AIR BASE NUMBER
*  2 * PERSONNEL TYPE
*  3 * NUMBER OF POINTS (UP TO 20)
*  4 *      POINT 1
*    *      .
*    *      .
*    *      .
* 23 *      POINT 20
*      *
*****

```



```

*****
*
*      A I R   B A S E   E V E N T   R E C O R D S
*
*****
*      *
*  WORD *  DESCRIPTION
*      *
*****
*      *
*  1 *  AIR BASE NUMBER
*  2 *  TIME
*  3 *  CREW RATIO
*  4 *  ALERT RATIO
*  5 *  HOURS PER CREW PER WEEK
*      *
*****

```

```

*****
*
*           C C T S   R E C O R D S
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * AIR BASE NUMBER
*  2 * TIME
*  3 * COURSE NUMBER
*  4 * PERSONNEL TYPE
*  5 * RATIO
*      *
*****

```

```

*****
*
*           P M T   G R O U P   R E C O R D
*
*****
*
* WORD * DESCRIPTION
*
* 1 * AIR BASE NUMBER
* 2 * TIME
* 3 * PMT NUMBER
* 4 * PERIOD
*
*****

```

```

*****
*
*   P M T   C O U R S E   R E C O R D S
*
*****
*   *
* WORD * DESCRIPTION
*   *
*****
*   *
* 1 * AIR BASE NUMBER
* 2 * TIME
* 3 * PMT NUMBER
* 4 * COURSE NUMBER
* 5 * PERSONNEL TYPE
* 6 * RATIO
* 7 * TIME LOST
*   *
*****

```



```

*****
*
* A I R   C R A F T   D E L I V E R Y   R E C O R D S
*
*****
*      *
* W O R D *   D E S C R I P T I O N
*      *
*****
*      *
*   1 *   A I R   B A S E   N U M B E R
*   2 *   T I M E
*   3 *   Q U A N T I T Y
*      *
*****

```



```

*****
*
*           S O U R C E   R E C O R D S
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*      *
*  1 * SOURCE NUMBER
*  2 * START TIME
*  3 * END TIME
*  4 * GENERATING FUNCTION NUMBER
*  5 * NUMBER OF PARAMETERS (UP TO 5)
*  6 *   PARAMETER 1
*   *   .
*   *   .
*   *   .
* 10 *   PARAMETER 5
*      *
*****

```

```

*****
*
*           R E S O U R C E   R E C O R D S
*
*****
*      *
* WORD * DESCRIPTION
*      *
*****
*  1 * RESOURCE NUMBER
*  2 * START TIME
*  3 * END TIME
*  4 * GENERATING FUNCTION NUMBER
*  5 * NUMBER OF PARAMETERS (UP TO 5)
*  6 *     PARAMETER 1
*  . *     .
*  . *     .
*  . *     .
* 10 *     PARAMETER 5
*      *
*****

```

## Section 1.12

### PHASE 1 ERROR MESSAGES

#### ADJVE CARD IS OUT OF SEQUENCE

A card which requires a header card to precede it, was encountered before the header card. (From INPUT)

#### ERROR AT CARD NUMBER XX, BLOCK NUMBER SYNCHRONIZED TO IS INVALID - YY

A processing block card specifies a synchronize or correlate reference to another block number which does not exist in the specified course. The card sequence number of the error is given by XX, and the invalid block number is given by YY. (From PROCB2)

#### ERROR AT CARD NUMBER XX, INVALID TRANSFER BLOCK NUMBER - YY

The processing block specified by card number XX specifies a transfer from a processing block which was never defined within that course. The invalid block number is given by YY. (From PROCB2)

#### ERROR IN SUBROUTINE IPLOT - INSUFFICIENT STORAGE AVAILABLE TO DO BLOCK DIAGRAM PLOT

The quantity of inputs was great enough so that there is not enough storage left for the plot routines work areas. The program will continue, but no plot will be produced. (From IPLOT)

#### ERROR IN SUBROUTINE PLOTB - BLOCK NUMBER XX WAS ENCOUNTERED BEFORE ANY BLOCK SPECIFYING A TRANSFER FROM IT

The processing blocks are out of sequence. The position of each processing block is specified as an offset from the block to the right of it (toward graduation). Therefore, each time a processing block is specified, another block must have already specified a transfer from it. (From PLOTB) .

ERROR IN SUBROUTINE PLOTB - INSUFFICIENT WORKING STORAGE AVAILABLE -  
FLOW ARROWS WILL BE OMITTED

The course is structured so that many processing blocks specify transfers from block numbers which are not defined. This message is printed when the plot routine runs out of room to store the references until they are defined. Usually the processing blocks can be specified in a different order to reduce the number of such references, but if no, the program will have to be recompiled to make storage available. (From PLOTB)

ERROR ON CARD NUMBER XX, BLOCK NUMBER YY HAS BEEN PREVIOUSLY DEFINED

Two processing blocks with the same number have been defined within the same course. (From PROCB1)

ERROR ON CARD NUMBER XX, DATA BLOCK NAME PREVIOUSLY DEFINED - YY

Card number XX attempts to define a data block with the name YY, but the same name has already been used for another block. (From NAME1)

ERROR ON CARD NUMBER XX, INVALID REFERENCE - YY

The card has referenced another data block which was never defined. The undefined block name or processing block number is given by YY. (From REPLC1, REPLC2)

INSUFFICIENT STORAGE AVAILABLE FOR INPUTS

The amount of input data is greater than the amount the program can store. The program will have to be re-compiled with more storage made available to it. (From INPUT)

INSUFFICIENT STORAGE AVAILABLE TO CONSTRUCT BLOCK NAME TABLE

The quantity of input data is large enough so that there is not enough storage available to do the cross referencing of data block names. The program will have to be re-compiled to make more storage available. (From NAME1, MAIN)



#### INVALID CARD NAME ON ABOVE CARD

This message appears in the input card listing. The card printed immediately above the message has a card name which is not recognized by the program (card name field is columns 1-10.) (From INPUT)

#### INVALID VALUE IN FIELD NUMBER XX

The card printed immediately above this error message contains a numeric value which is outside the range allowed for that value. The field number XX, refers to the field number marking at the top of the input card listing. (From TEST)

#### RESOURCE NAME MUST NOT BE BLANK

The RUDB card which is printed above this error message does not have a resource name specified. (From INPUT)

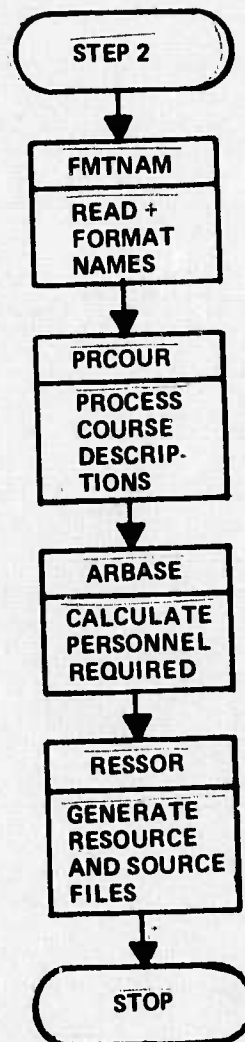


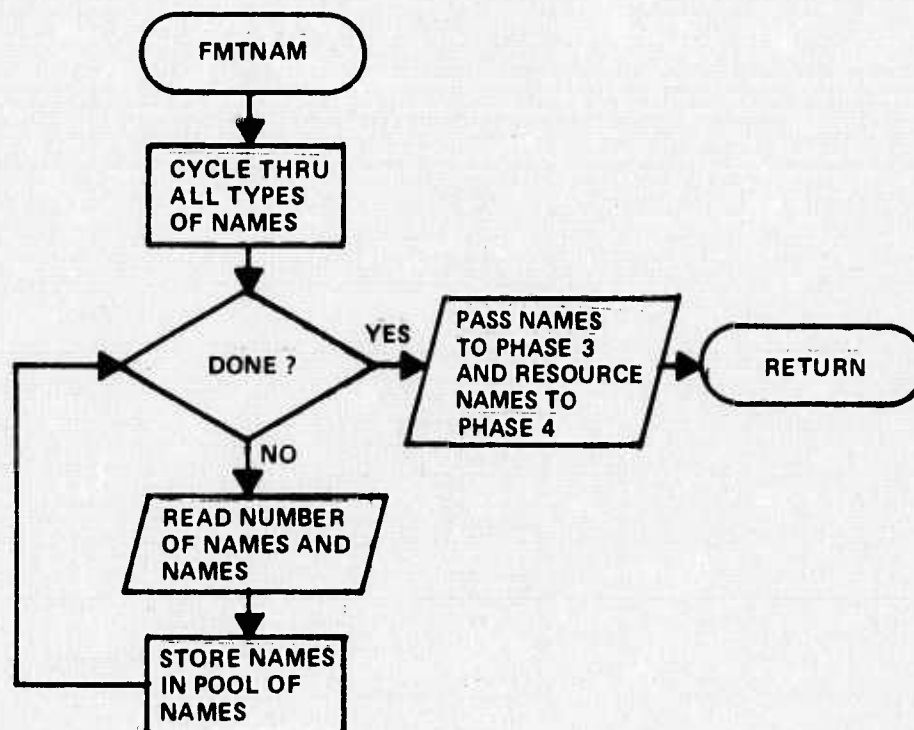
## Section 2.0

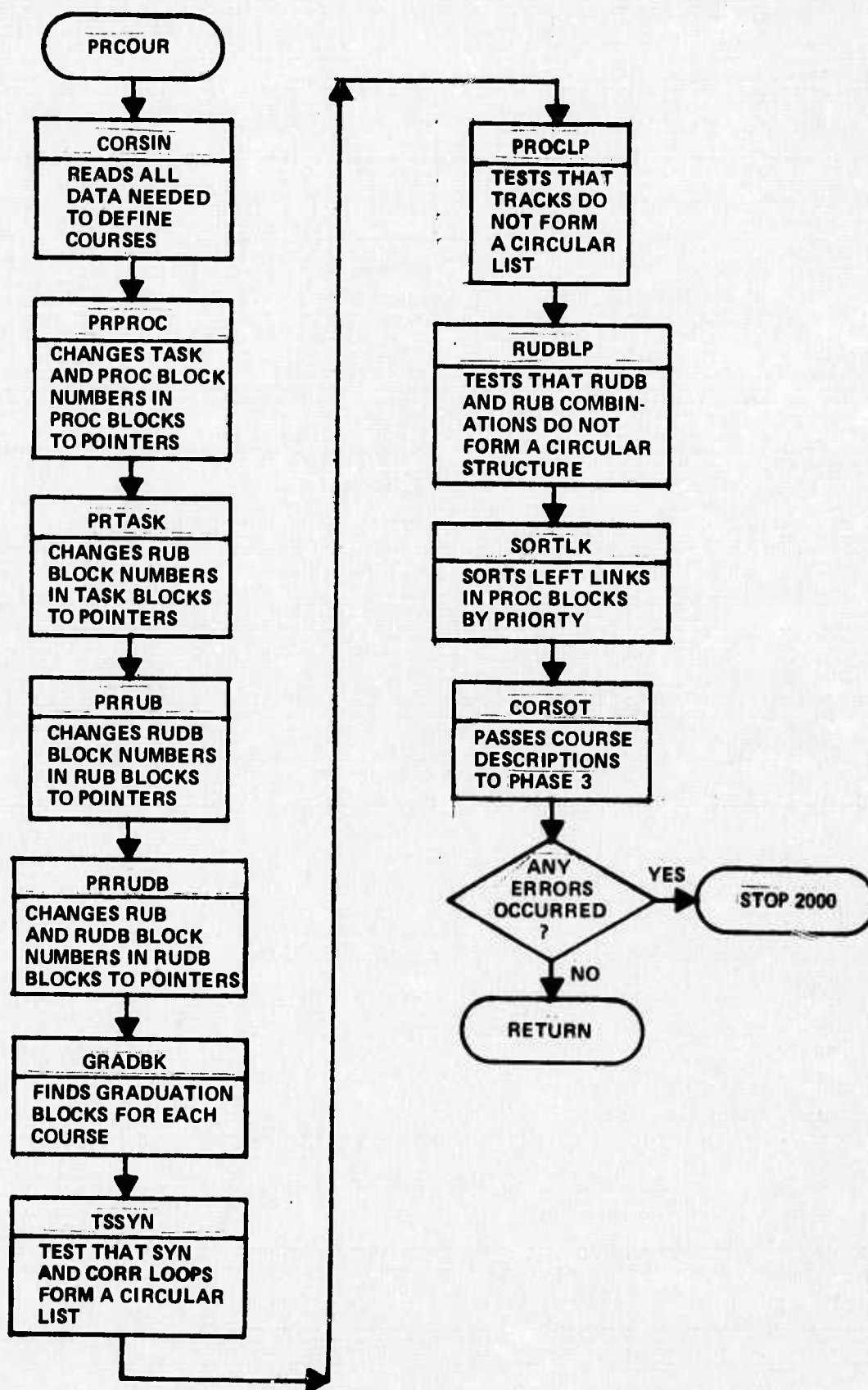
### TRAM PHASE 2

This section provides flowcharts, record formats, common block description, subroutine description and a symbol cross reference for Phase 2. This information is intended to supplement the description included in Technical Memorandum SAT-5, TRAM User's Manual with which the reader is assumed to be familiar.

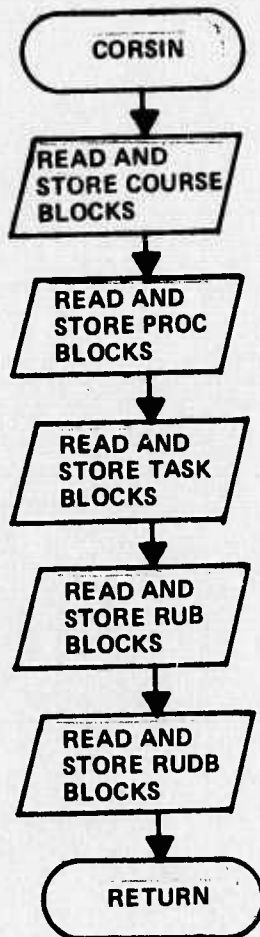
Section 2.1  
FLOWCHARTS

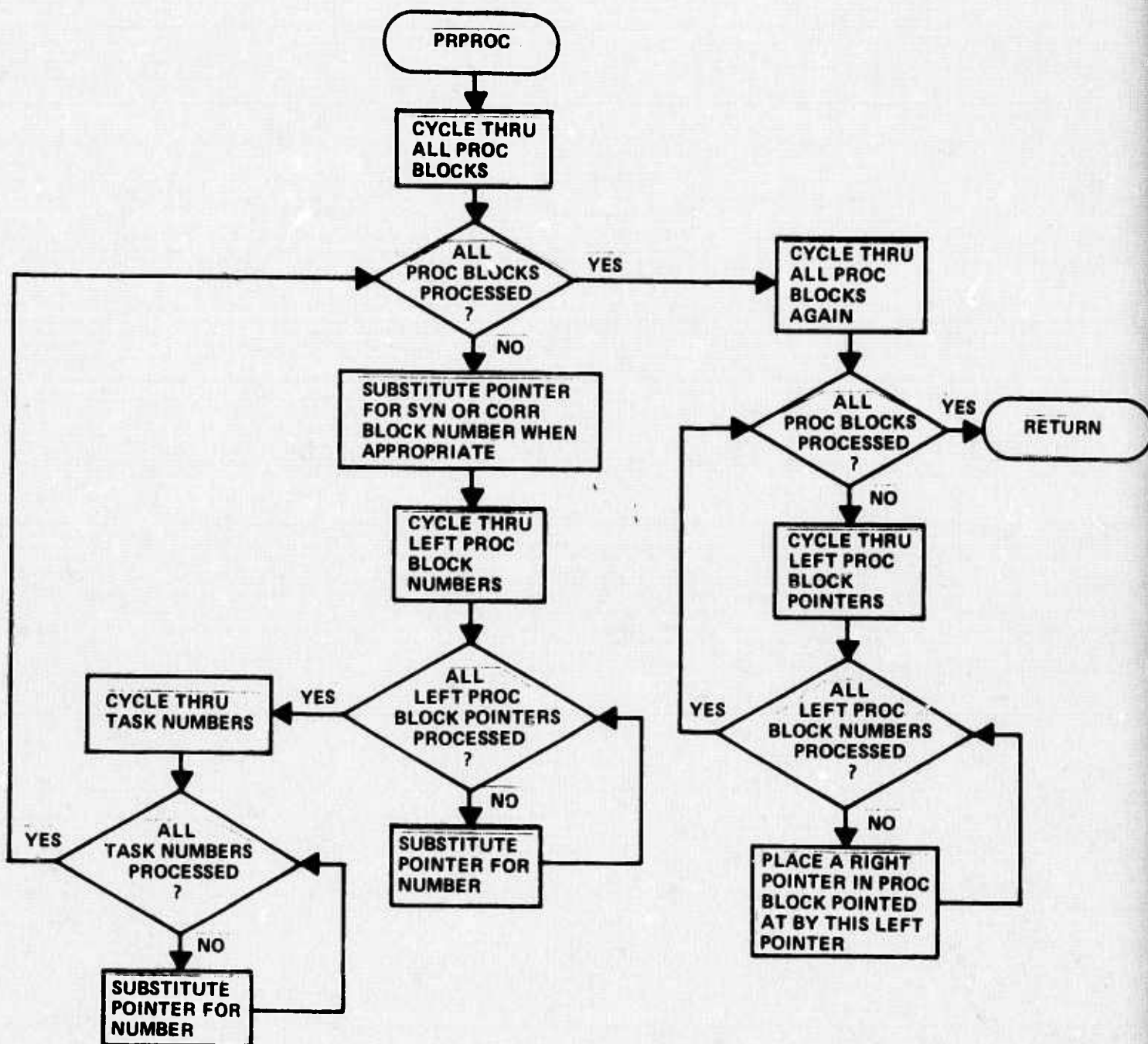


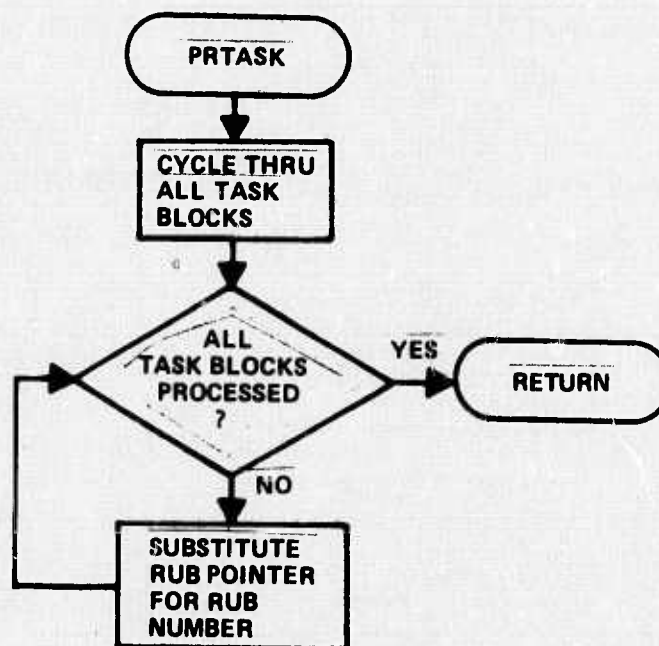


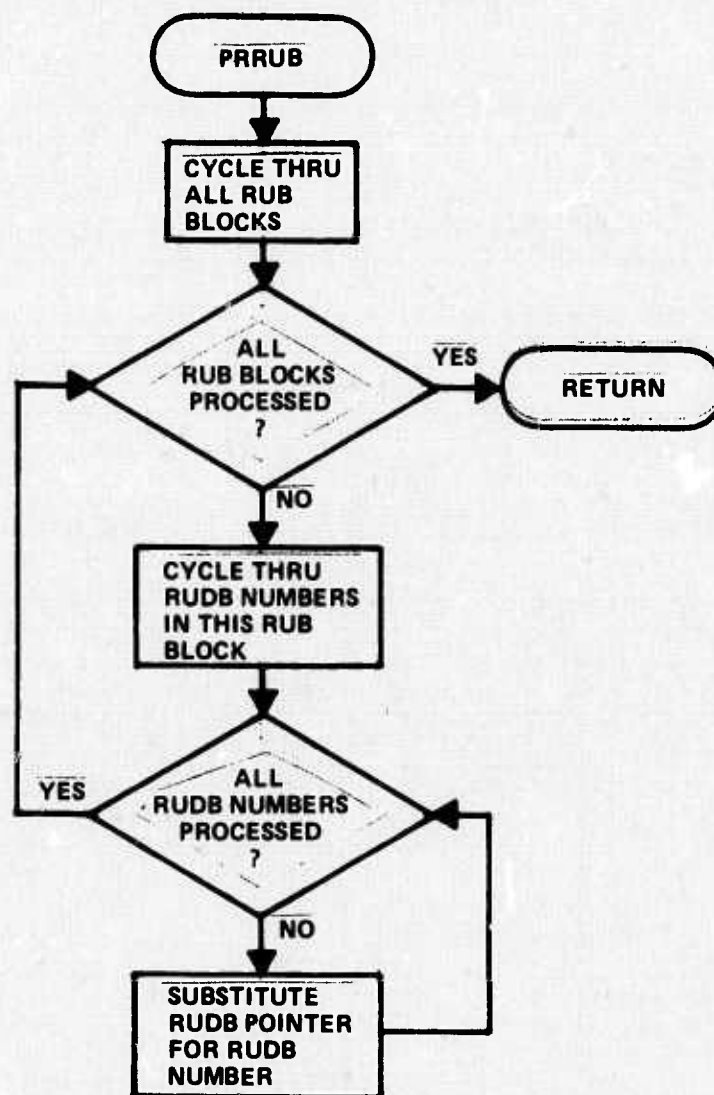




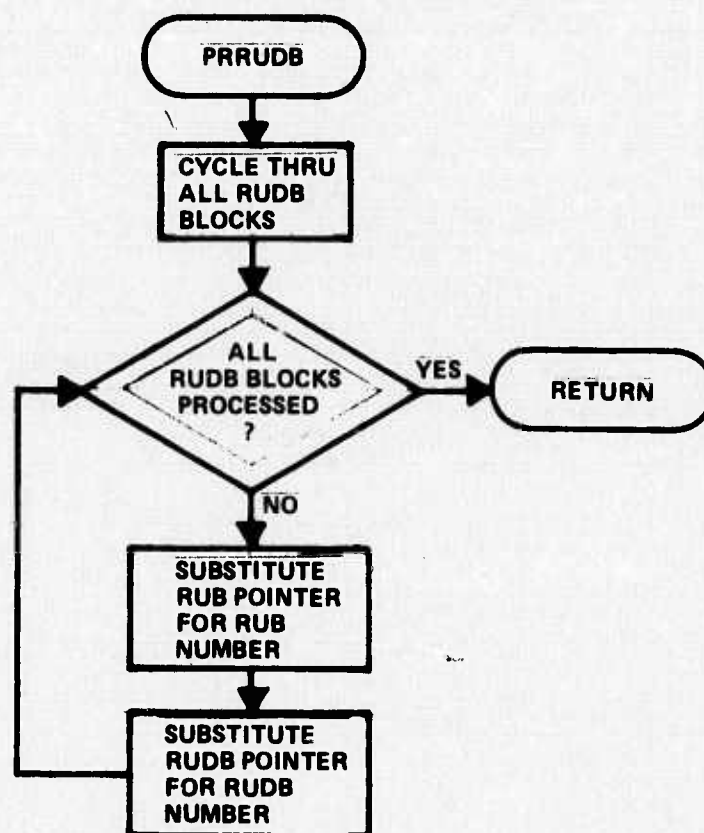




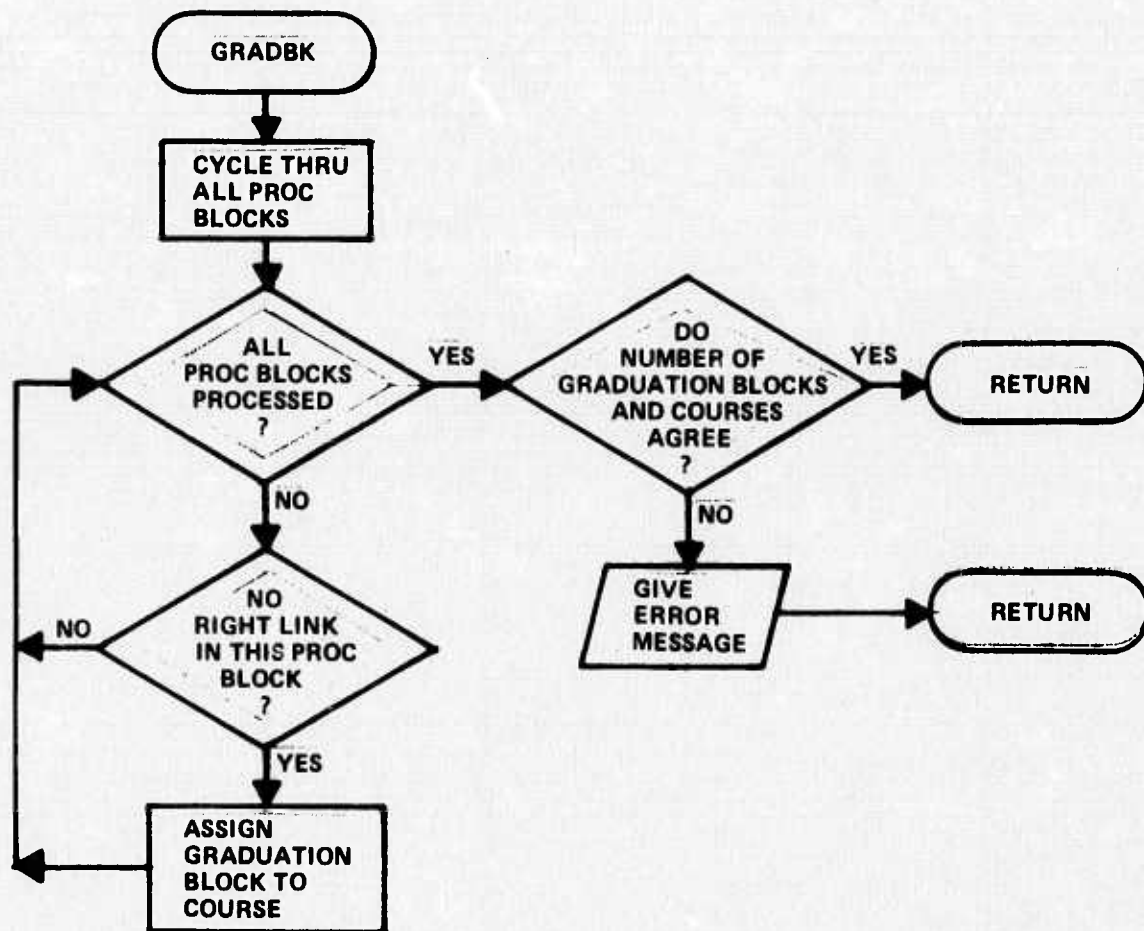


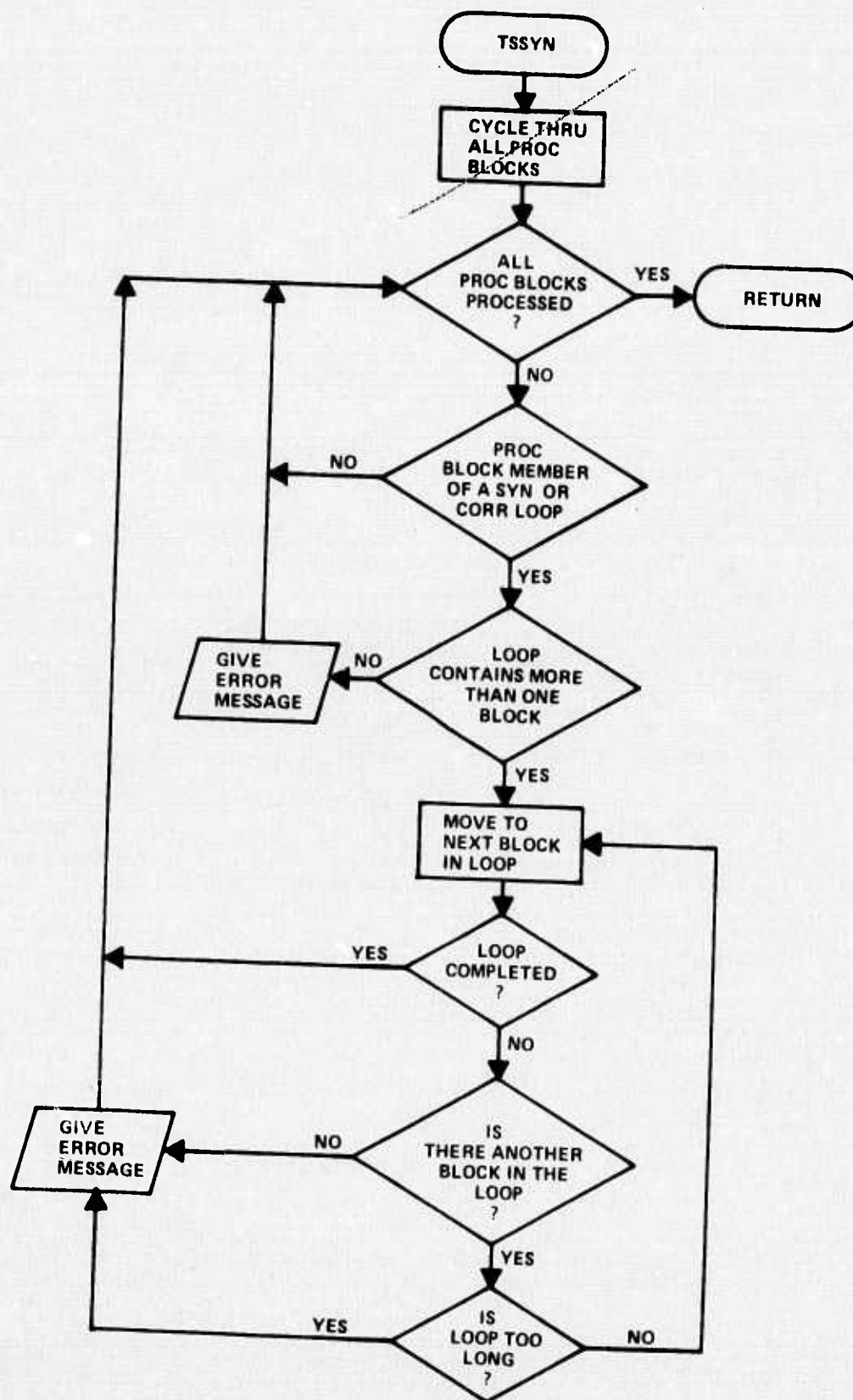


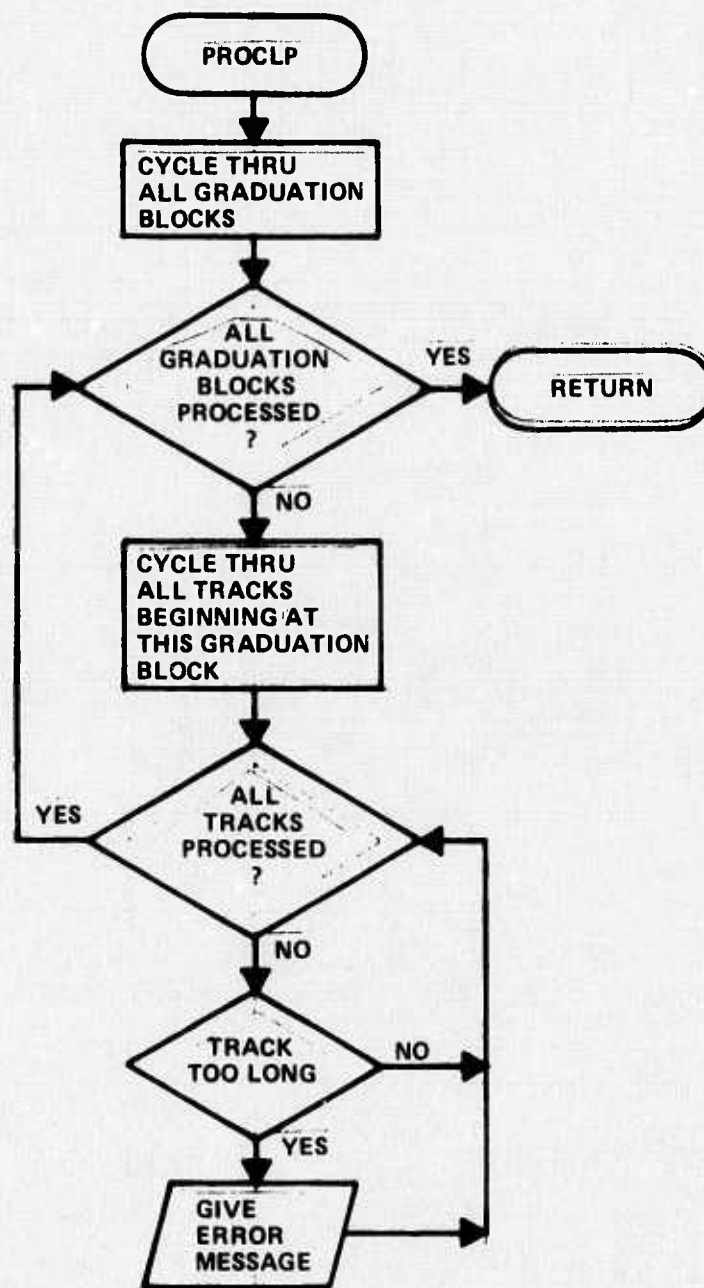


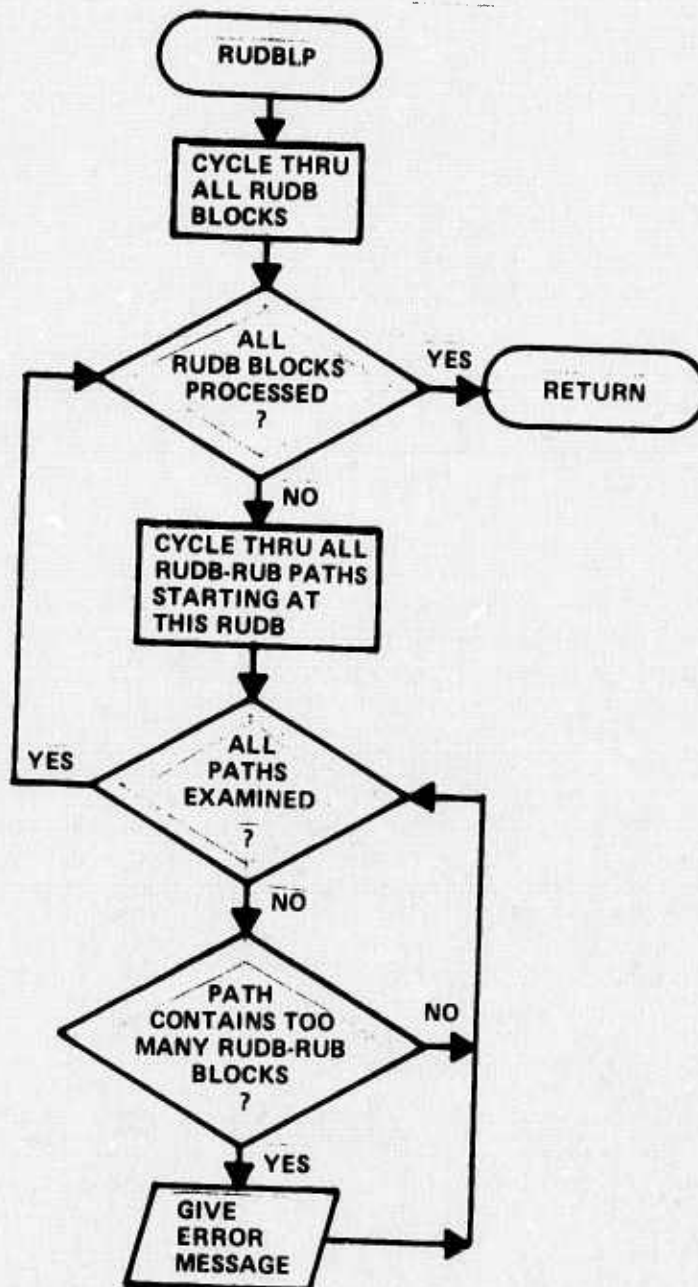




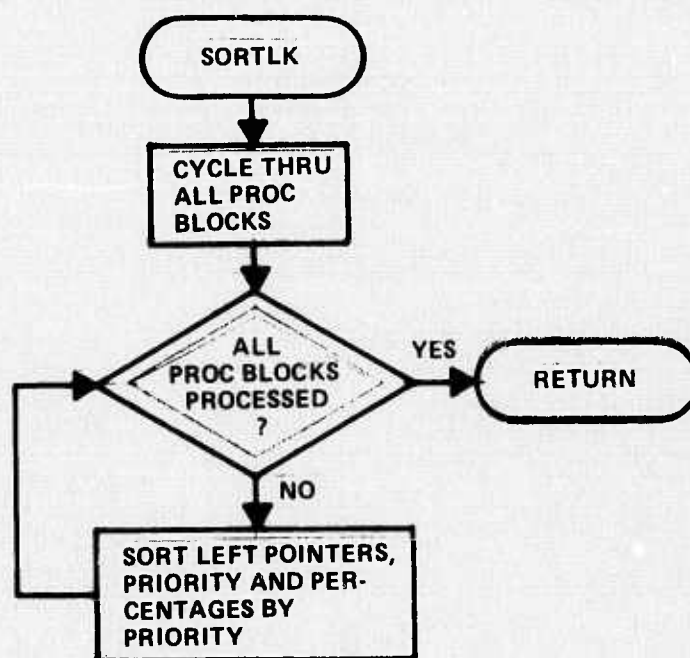




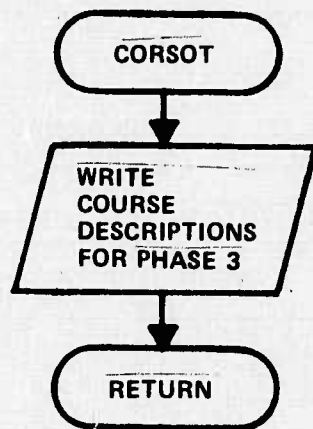


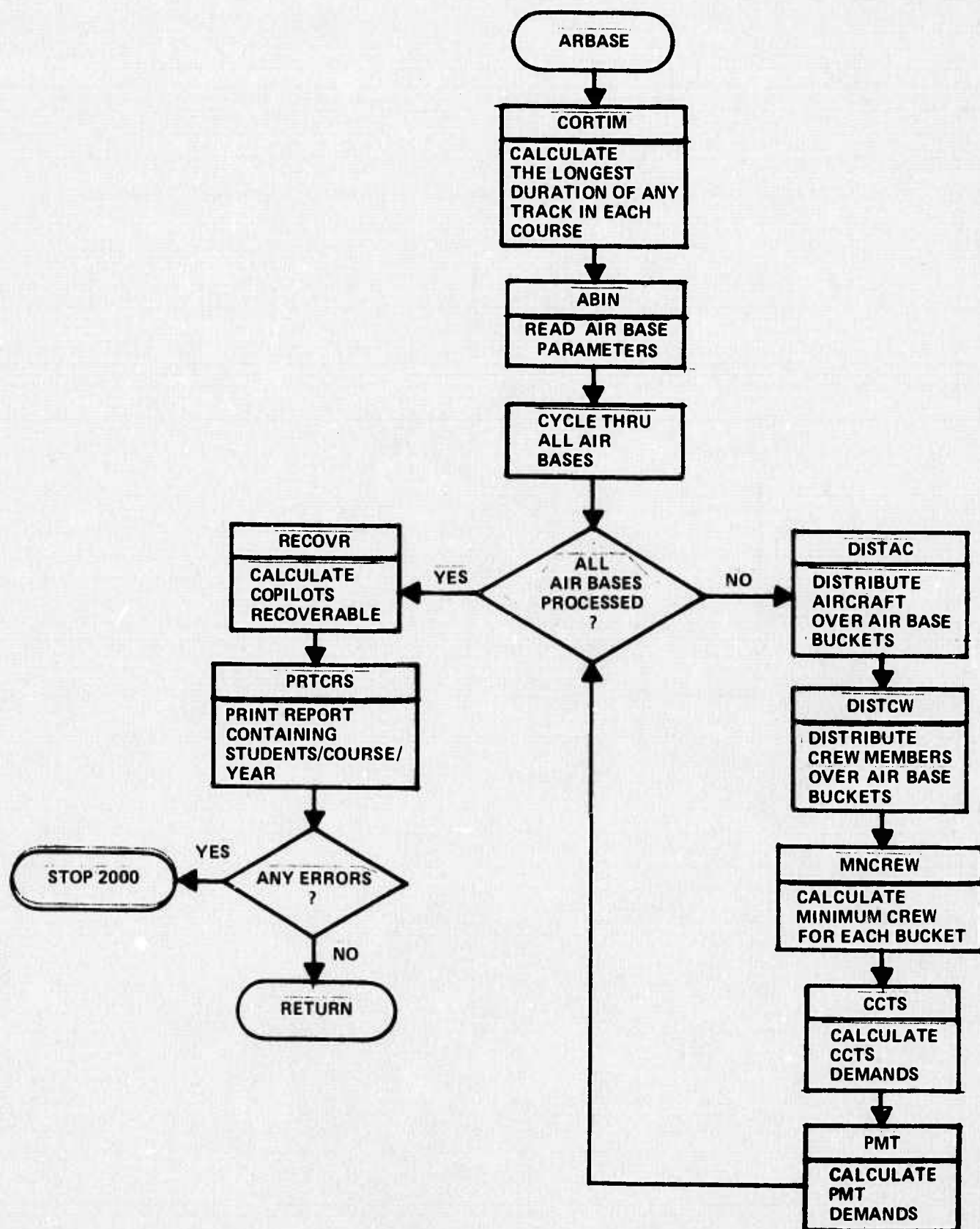


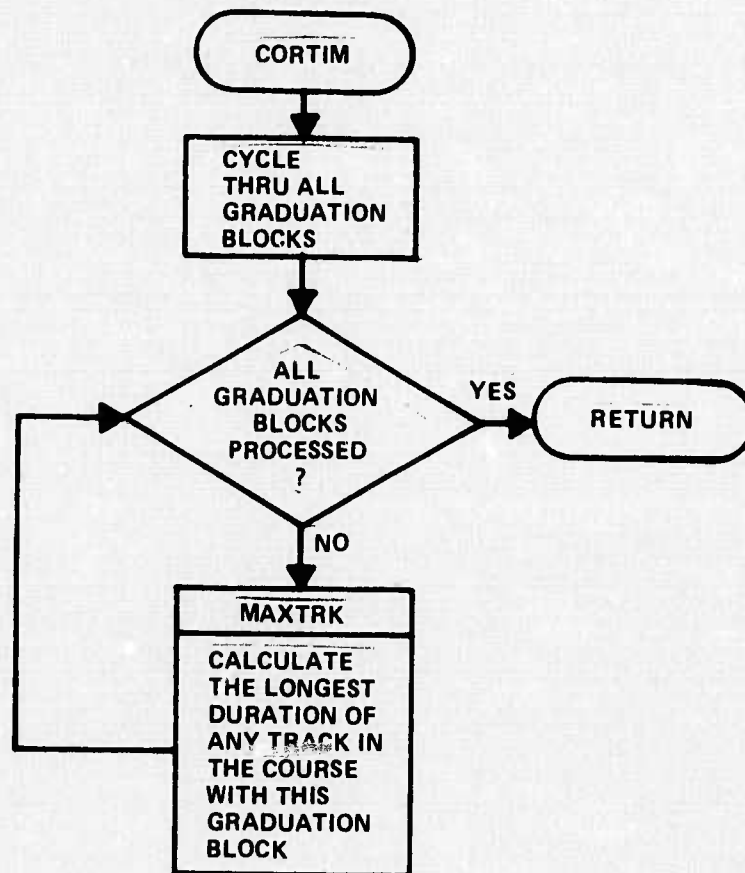


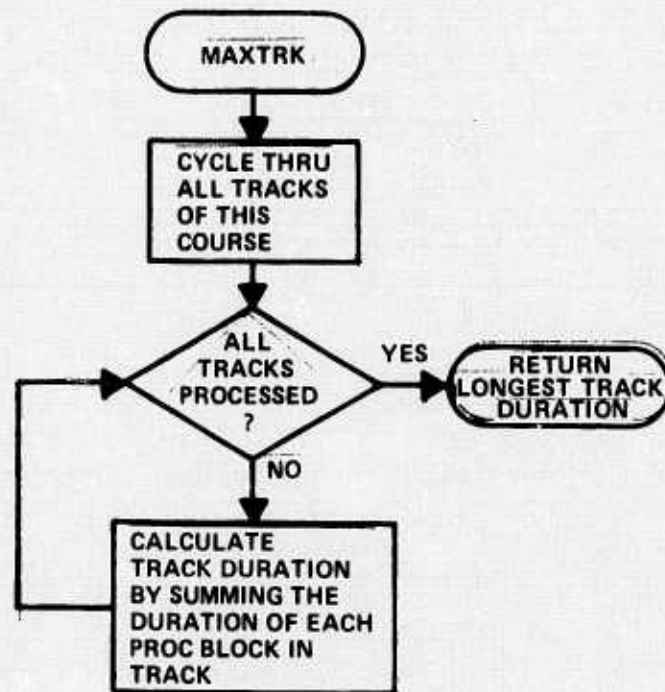




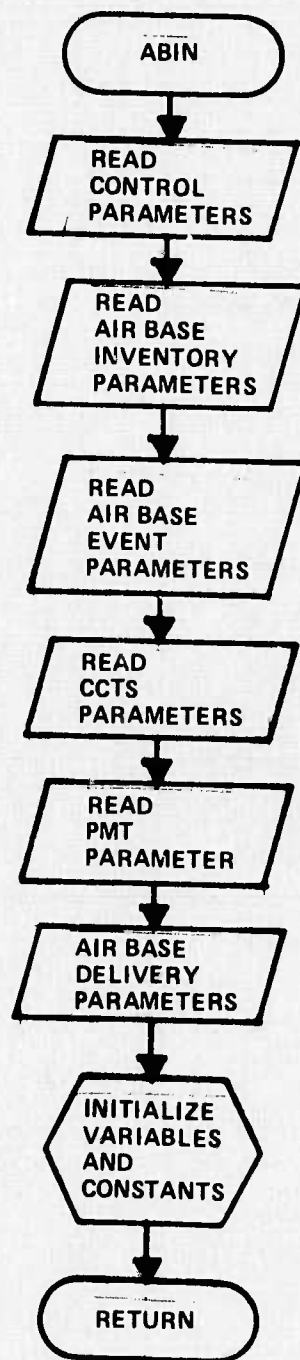




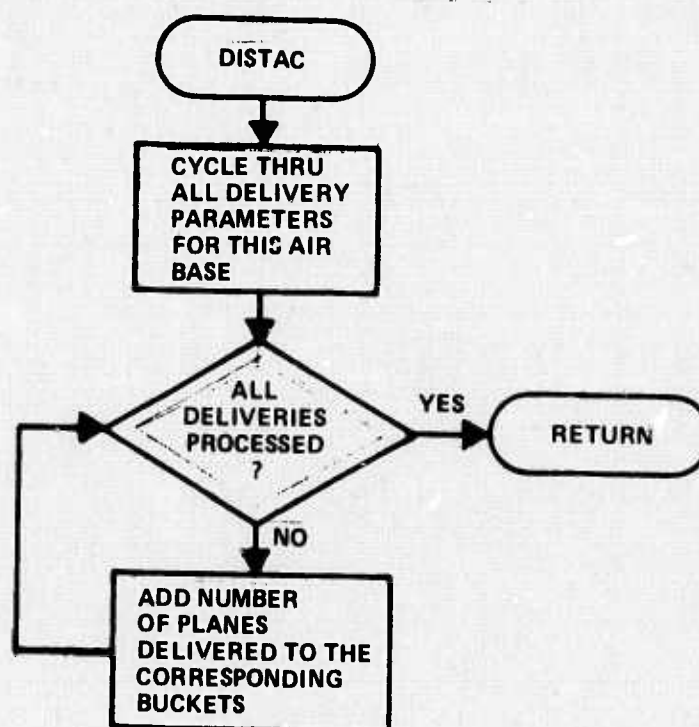


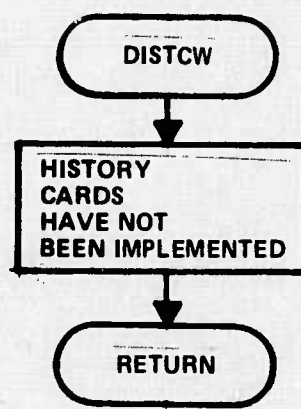


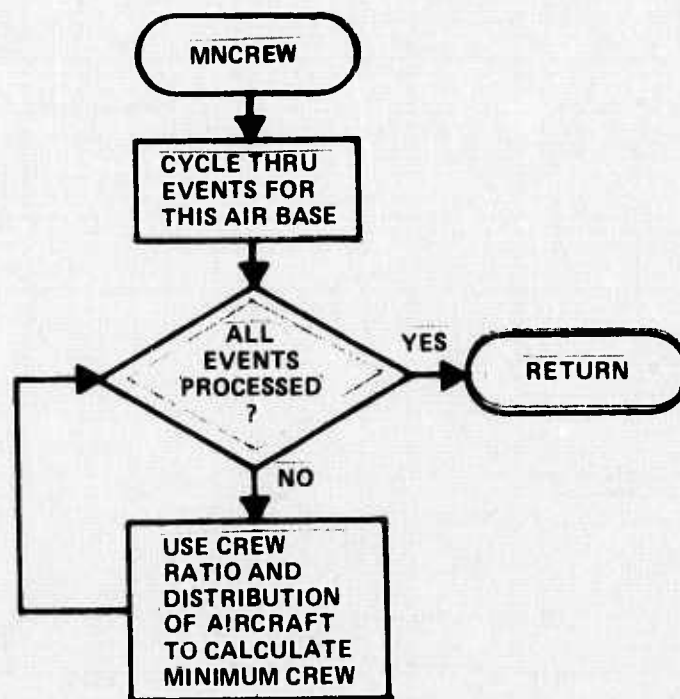


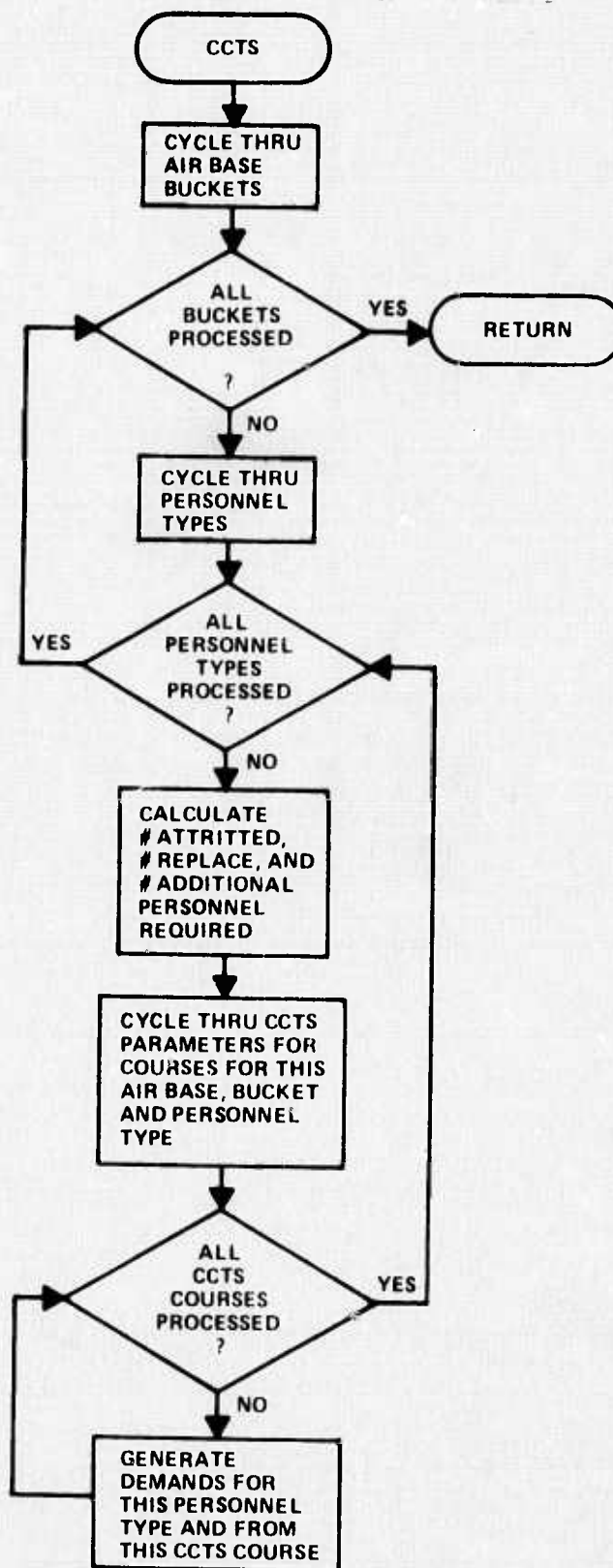




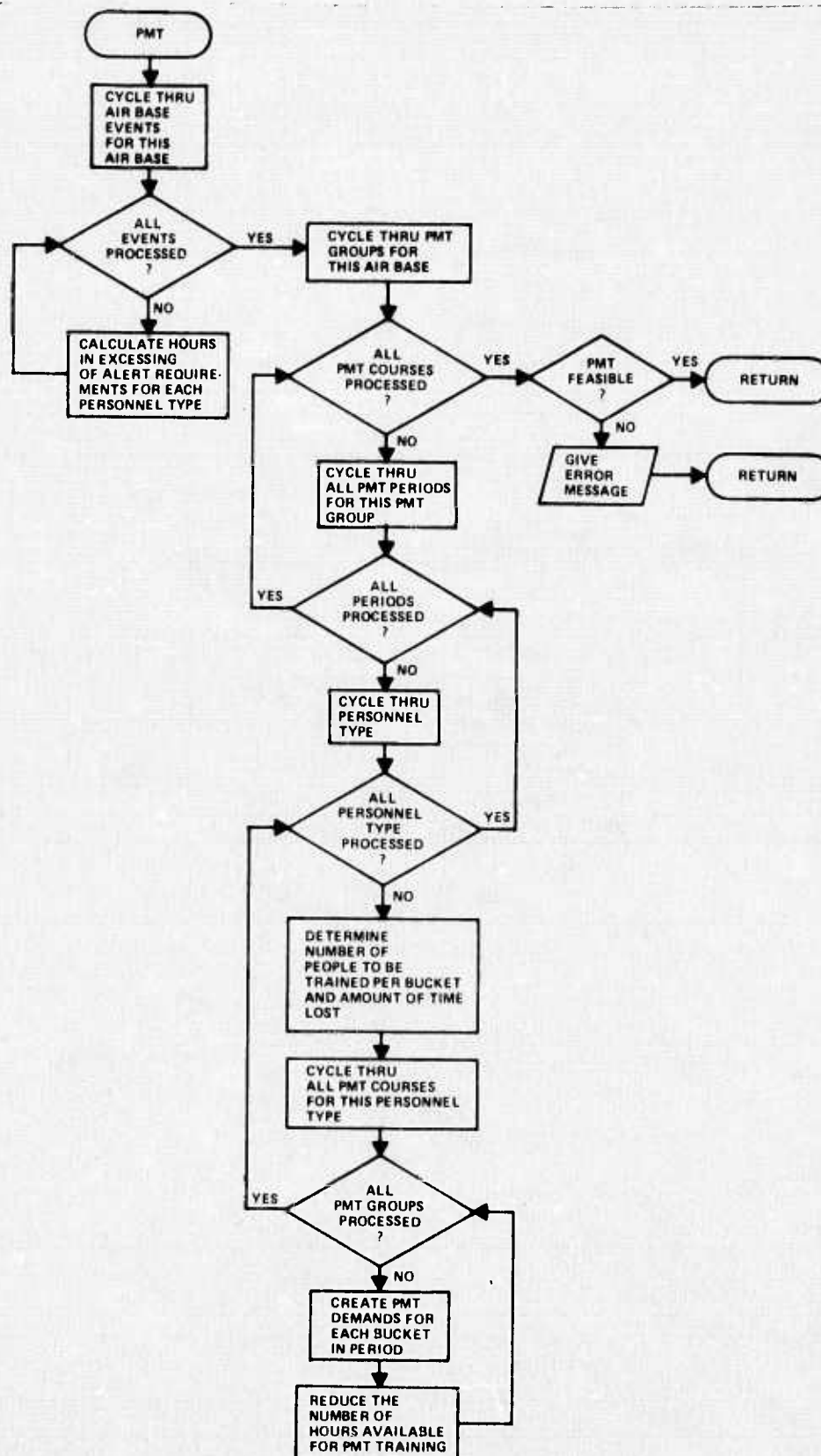




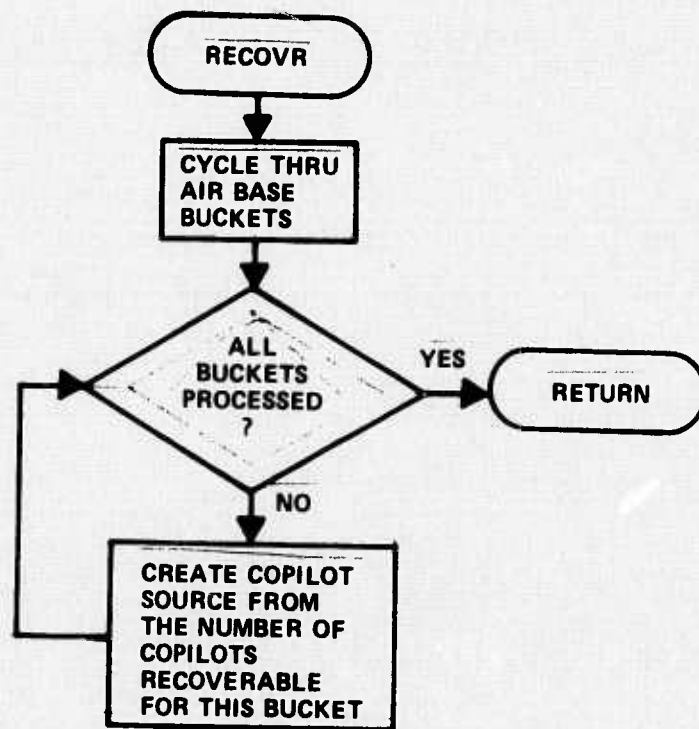


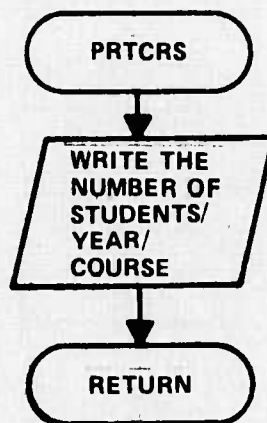


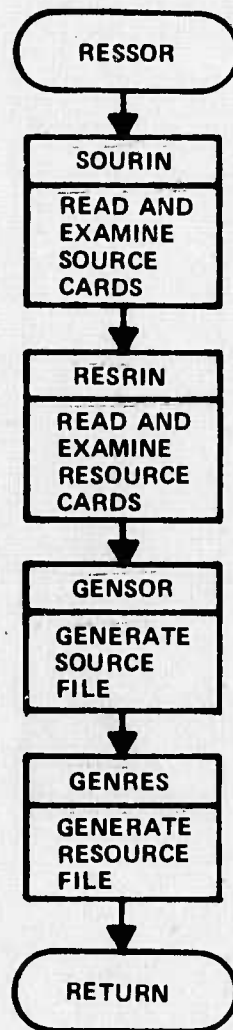


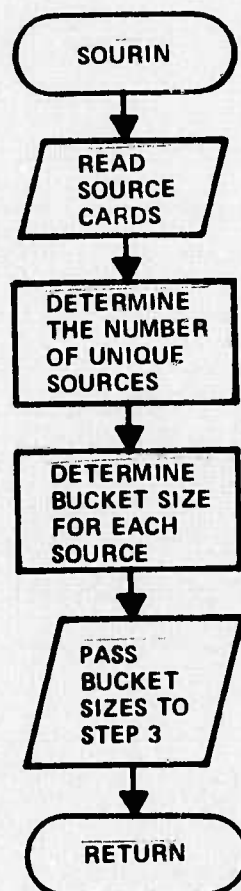




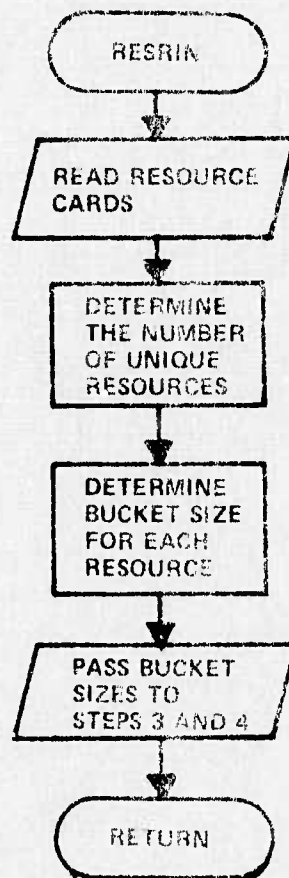




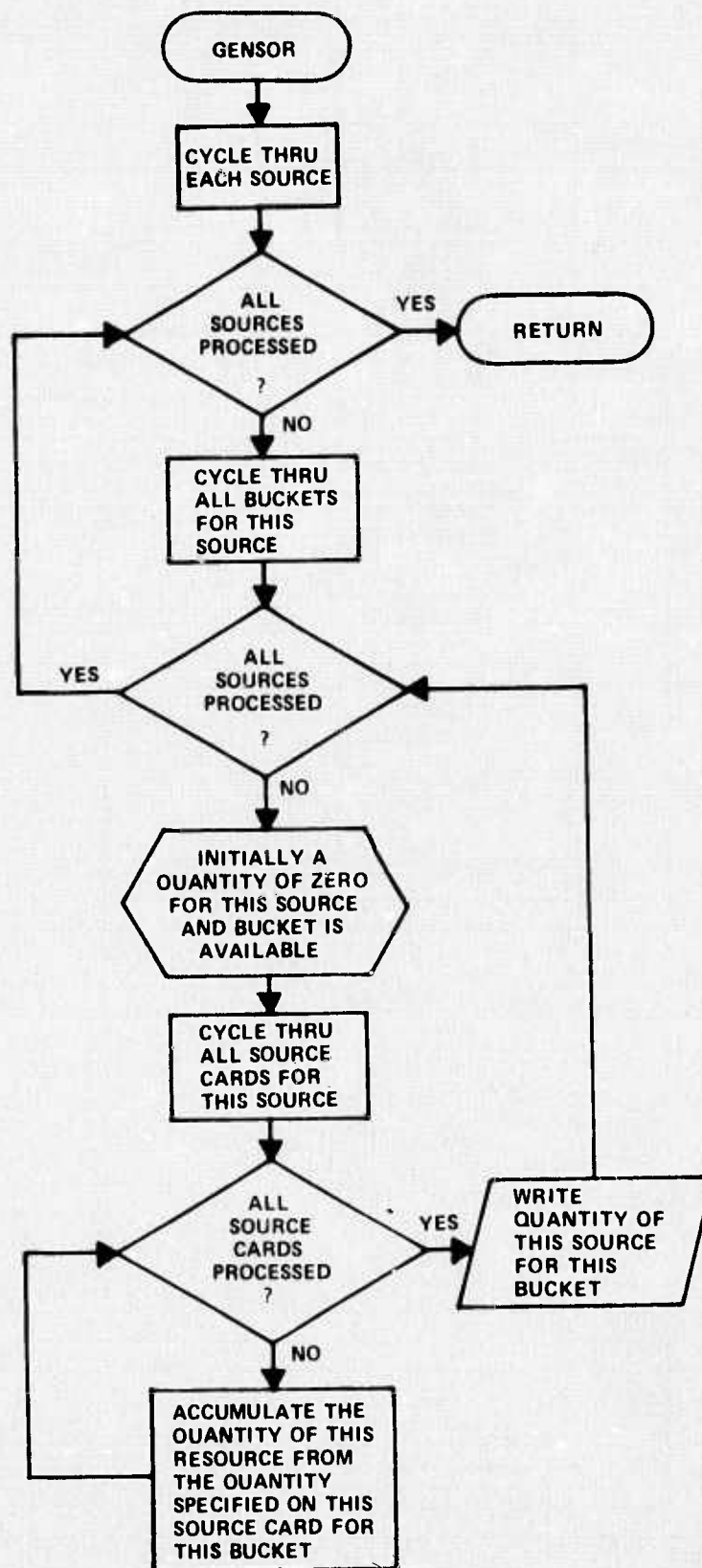


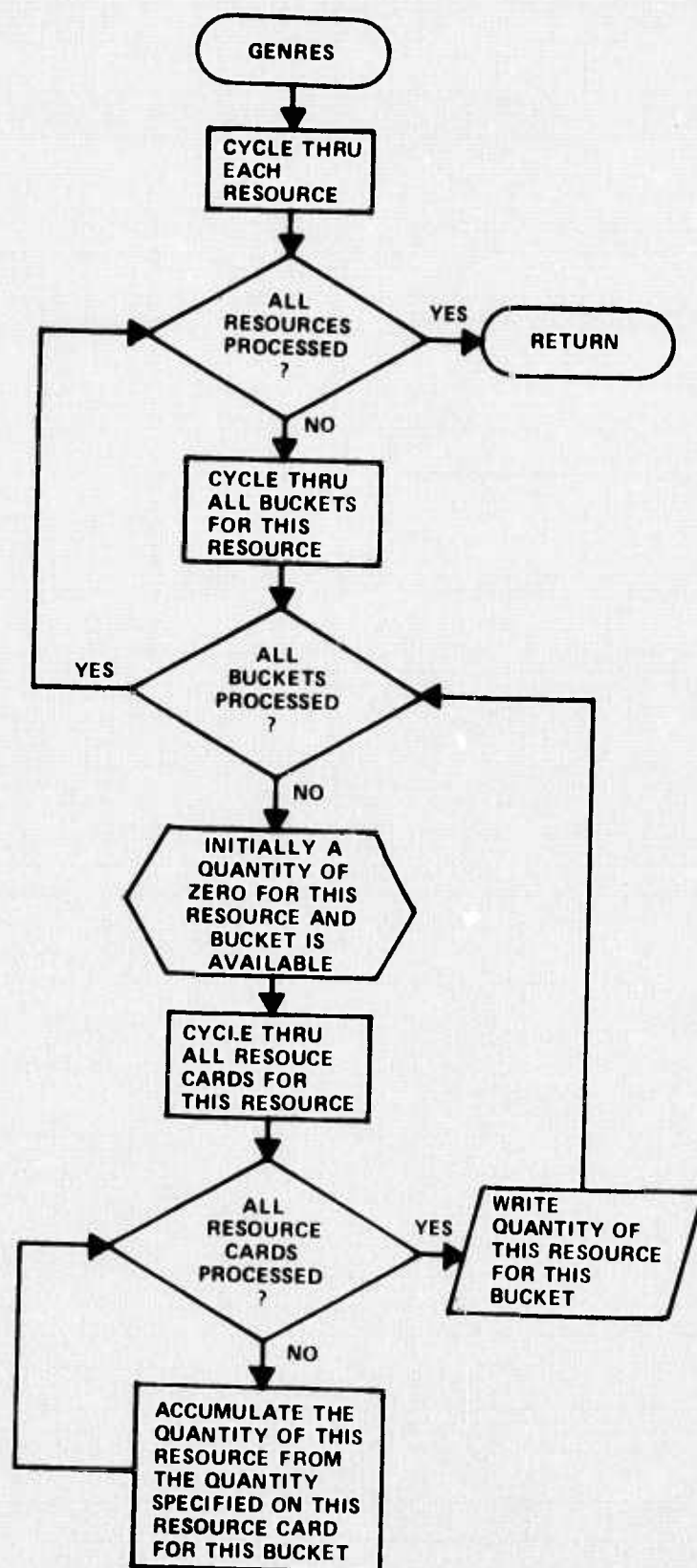


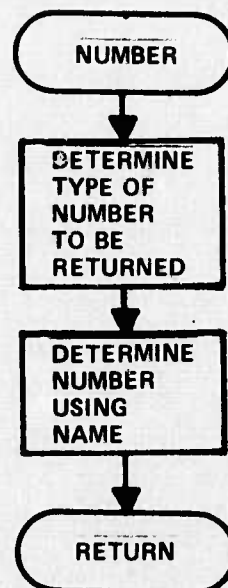


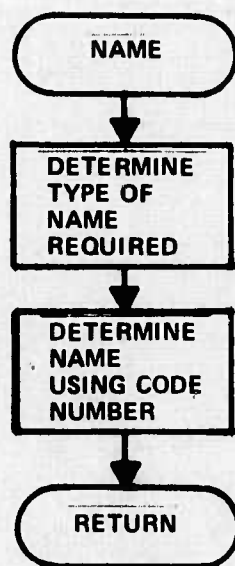














Section 2.2

DESCRIPTIONS OF RECORDS AND VARIABLES  
USED IN COMMONS



FIGURE C.1

```

*****
*
*   D E M A N D   R E C O R D
*
*****
*   *
* WORD *   D E S C R I P T I O N
*   *
*****
*   *
*   1 *   TIME
*   2 *   QUANTITY
*   3 *   TYPE OF PERSONNEL  1-PILOTS
*   *                           2-COPILOTS
*   *                           3-OSOS
*   *                           4-DSOS
*   4 *   COURSE NUMBER
*   5 *   DEMAND NUMBER = AIR BASE NUMBER * 1000 + BUCKET NO.
*   6 *   DEMAND TYPE  1-CCTS BECAUSE OF DELIVERIES
*   *                   2-CCTS BECAUSE OF ATTRITION
*   *                   3-PMT
*
*****

```

FIGURE C.2

```
*****
*
*   S O U R C E   R E C O R D
*
*****
*   *
* WORD *   D E S C R I P T I O N
*   *
*****
*   *
*   1 *   T I M E
*   2 *   S O U R C E   N U M B E R
*   3 *   Q U A N T I T Y
*   *
*****
```

FIGURE C.3

```
*****
*
*      R E S O U R C E   R E C O R D
*
*****
*      *
* WORD  *   D E S C R I P T I O N
*      *
*****
*      *
*      1 *   TIME
*      2 *   RESOURCE NUMBER
*      3 *   QUANTITY
*      *
*****
```

FIGURE D.1

```

*****
*
*          C O M M O N   B L O C K   -   N A M
*
*****
*
*  VARIABLE   *   D E S C R I P T I O N
*
*****
*
*  ITYPE(I)   *   TYPE OF NAME
*  IFIRST(I)  *   INDEX TO WHERE FIRST NAME OF TYPE I IS LOCATED
*  NUM(I)     *   NUMBER OF NAMES OF TYPE I
*  NAMES(I,J) *   POOL OF NAMES
*  IUNIT      *   UNIT NAMES ARE READ ON
*  JUNIT      *   UNIT NAMES ARE WRITTEN ONTO
*  MAXNUM     *   MAXIMUM NUMBER OF NAMES
*  NTYPE      *   NUMBER OF NAME TYPES
*
*****

```



FIGURE D.2

```

*****
*
*           C O M M O N   B L O C K   -   D U M M Y (COURSES)
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  NCOURS      *  NUMBER OF COURSES
*  MXCOUR      *  MAXIMUM NUMBER OF COURSES
*  IGRAD(1)    *  GRADUATION BLOCK FOR COURSE 1
*  ICTYPE(1)   *  TYPE OF COURSE 1
*  IPTYPE(1)   *  PERSONNEL TYPE FOR COURSE 1
*  IPRIOR(1)   *  PRIORITY OF COURSE 1
*  MXSIZE(1)   *  MAXIMUM SIZE OF COURSE 1
*  IPEROD(1)   *  PERIOD OF COURSE 1
*  IEGRAD(1)   *  EARLIEST GRADUATION DATE FOR COURSE 1
*  IBLOCK(1)   *  LOCATION OF FIRST PROC, TASK, RUB, AND RUDB BLOCK
*  NBLOCK(1)   *  NUMBER OF PROC, TASK, RUB, AND RUDB BLOCKS
*  LBKIN(1)    *  LENGTH OF EACH BLOCK TYPE AS READ FROM STEP1
*  LBKOUT(1)   *  LENGTH OF EACH BLOCK TYPE AS PASSED TO STEP3
*  IAVAIL      *  POINTS TO NEXT AVAILABLE WORD IN STORAGE POOL
*  NWORDS      *  NUMBER OF WORDS REMAINING IN STORAGE POOL
*  IWORDS(1)   *  POOL OF STORAGE FOR BLOCKS
*  ERROR       *  TRUE IFF AN ERROR OCCURRED
*
*****

```



FIGURE D.3

***** * * COMMON BLOCK - DUMMY (AIR BASES) * *****	
* VARIABLE	* DESCRIPTION
*****	
* MAB	* MAXIMUM NUMBER OF AIR BASES
* MABH	* MAXIMUM NUMBER OF HISTORY CARDS
* MABE	* MAXIMUM NUMBER OF AIR BASE EVENTS
* MABC	* MAXIMUM NUMBER OF CCTS COURSES
* MABP	* MAXIMUM NUMBER OF PMT GROUPS
* MABPC	* MAXIMUM NUMBER OF PMT COURSES
* MABD	* MAXIMUM NUMBER OF DELIVERY CARDS
* MBUCKT	* MAXIMUM NUMBER OF AIR BASE BUCKETS
* PPATTR	* % PILOTS ATTRITION PER BUCKET
* CPATTR	* % COPILOTS ATTRITION PER BUCKET
* OPATTR	* % USOS ATTRITION PER BUCKET
* DPATTR	* % DSOS ATTRITION PER BUCKET
* IPATTD	* PILOTS ATTRITION DELAY TIME
* ICATTD	* COPILOTS ATTRITION DELAY TIME
* IOATTD	* OSOS ATTRITION DELAY TIME
* IDATTD	* DSOS ATTRITION DELAY TIME
* PCRECY	* % COPILOTS RECOVERABLE
* ICUYR	* CALENDAR UNITS PER YEAR
* IBUCKT	* CALENDAR UNITS PER BUCKET
* WKS BKT	* WEEKS PER BUCKET
* HRSCU	* HOURS PER CALENDAR UNIT
* HRBUCK	* HOURS PER BUCKET
* MXBUCK	* BUCKET ASSOCIATED WITH MAXIMUM SIMULATION TIME
* NAB	* NUMBER OF AIR BASES
* INVAC(I)	* INITIAL AIR CRAFT INVENTORY FOR AIR BASE I
* INVPI(I)	* INITIAL PILOT INVENTORY FOR AIR BASE I
* INVC(I)	* INITIAL COPILOT INVENTORY FOR AIR BASE I
* INVO(I)	* INITIAL OSO INVENTORY FOR AIR BASE I
* INVD(I)	* INITIAL DSO INVENTORY FOR AIR BASE I
* NABH	* NUMBER OF AIR BASE HISTORY CARDS
* NABE	* NUMBER OF AIR BASE EVENTS
* IABE1(I)	* INDEX OF FIRST AIR BASE EVENT FOR AIR BASE I
* IABEN(I)	* NUMBER OF AIR BASE EVENTS FOR AIR BASE I
* IDATEE(I)	* DATE OF AIR BASE EVENT (IN BUCKETS)
* CREWR(I)	* CREW RATIO FOR AIR BASE EVENT
* ALERTR(I)	* ALERT RATIO FOR AIR BASE EVENT
* HRCRBK(I)	* HOURS/CREW/BUCKET FOR AIR BASE EVENT
* NABC	* NUMBER OF CCTS COURSES
* IABC1(I)	* INDEX OF FIRST CCTS FOR AIR BASE I
* IABCN(I)	* NUMBER OF CCTS FOR AIR BASE I
* IDATEC(I)	* DATE OF CCTS (IN BUCKETS)
* ICOURC(I)	* COURSE NUMBER
* IPERC(I)	* PERSONNEL TYPE
* PCC(I)	* PER CENTAGE OF PEOPLE TO COME FROM THIS COURSE
*****	

FIGURE D.3 (CONTINUED)

```

*****
*
*          C O M M U N   B L O C K   -   D U M M Y (AIR BASES)
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NABP          *  NUMBER OF PMT GROUPS
*  IABP1(I)      *  INDEX OF FIRST PMT GROUP FOR AIR BASE 1
*  IABPN(I)      *  NUMBER OF FIRST PMT GROUP FOR AIR BASE 1
*  IDATEP(I)     *  DATE OF PMT GROUP (IN BUCKETS)
*  IPMT(I)       *  PMT NUMBER
*  IPEROD(I)     *  PMT PERIOD
*  NABPC         *  NUMBER OF PMT COURSES
*  IABPC1(I)     *  INDEX OF FIRST PMT COURSES FOR AIR BASE 1
*  IABPCN(I)     *  NUMBER OF PMT COURSES FOR AIR BASE 1
*  IDATPC(I)     *  DATE OF PMT COURSE (IN BUCKETS)
*  JPMT(I)       *  NUMBER OF PMT FOR PMT COURSE
*  ICOUPC(I)     *  COURSE NUMBER
*  IPERTP(I)     *  PERSONNEL TYPE FOR PMT COURSE
*  PCPC(I)       *  PER CENT OF PERSONNEL TO GO TO THIS PMT COURSE
*  ITL(I)        *  TIME DELAY DUE TO TRAVEL FOR PMT COURSE
*  NABD          *  NUMBER OF AIR CRAFT DELIVERIES
*  IABD1(I)      *  INDEX OF FIRST DELIVERY FOR AIR BASE 1
*  IABDN(I)      *  NUMBER OF DELIVERY CARDS FOR AIR BASE 1
*  IDATED(I)     *  DATE OF DELIVERY
*  IQANTD(I)     *  QUANTITY DELIVERIED
*  P(I)          *  NUMBER OF PILOTS FOR BUCKET 1
*  C(I)          *  NUMBER OF COPILOTS FOR BUCKET 1
*  O(I)          *  NUMBER OF OSOS FOR BUCKET 1
*  D(I)          *  NUMBER OF DSOS FOR BUCKET 1
*  IAC(I)        *  NUMBER OF AIR CRAFT FOR BUCKET 1
*  CREW(I)       *  MINIMUM NUMBER OF CREWS FOR BUCKET 1
*  PH(I)         *  PILOT HOURS AVAILABLE FOR PMT FOR BUCKET 1
*  CH(I)         *  COPILOT HOURS AVAILABLE FOR PMT FOR BUCKET 1
*  OH(I)         *  OSC HOURS AVAILABLE FOR PMT FOR BUCKET 1
*  DH(I)         *  DSU HOURS AVAILABLE FOR PMT FOR BUCKET 1
*  CRECY(I)      *  COPILOTS RECOVERABLE FOR BUCKET 1
*  IAB           *  NUMBER OF AIR BASE BEING PROCESSED
*  ERROR         *  TRUE IFF ERROR OCCURRED
*  NOPRNT        *  TRUE IFF OPTION REPORT IS NOT TO BE PRINTED
*  STUJS(1,J)    *  # OF STUDENTS SENT TO COURSE J DURING YEAR 1
*  NCORS         *  TOTAL NUMBER OF COURSES
*  NYEARS        *  NUMBER OF YEARS OF SIMULATION TIME
*
*****

```

FIGURE D.4

```
*****
*
*          COMMON BLOCK - CONTROL (STEP2)
*
*****
*          *
* VARIABLE * DESCRIPTION
*          *
*****
*          *
* MXTIME  * MAXIMUM SIMULATION TIME
*          *
*****
```



FIGURE D.5

```
*****
*
*          COMMON BLOCK - RECVRY          *
*
*****
*
* VARIABLE      * DESCRIPTION              *
*
*****
* ICRECT      * # CALENDAR UNITS COPILOTS ARE AVAILABLE *
* ICBUCK      * # OF BUCKETS RECOVERED COPILOTS ARE AVAILABLE *
*
*****
```

FIGURE D.6

```
*****
*
*          COMMON BLOCK - MAXLEN          *
*
*****
*          *
* VARIABLE * DESCRIPTION                  *
*          *
*****
*          *
* MAXTIM(1) * DURATION OF LONGEST TRACK IN COURSE 1 *
*          *
*****
```



FIGURE D.7

```

*****
*
*          C O M M O N   B L O C K   -   R E S U R S
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IRES1(1)      *  INDEX TO FIRST RESOURCE CARD FOR RESOURCE 1
*  NRES(I)       *  NUMBER OF RESOURCE CARDS FOR RESOURCE 1
*  IFUNC(I)      *  GENERATING FUNCTION
*  IT1(I)        *  BEGINNING OF RESOURCE AVAILABILITY
*  ITN(I)        *  END OF RESOURCE AVAILABILITY
*  NPARM(I)      *  NUMBER OF PARAMETERS
*  IPARM1(I)     *  PARAMETER1
*  IPARM2(I)     *  PARAMETER2
*  IPARM3(I)     *  PARAMETER3
*  IPARM4(I)     *  PARAMETER4
*  IPARM5(I)     *  PARAMETER5
*  LBUCKET(I)    *  BUCKET SIZE
*  NRESR         *  TOTAL NUMBER OF RESOURCES
*  MXRES         *  MAXIMUM NUMBER OF RESOURCES
*  MXRESC        *  MAXIMUM NUMBER OF RESOURCE CARDS
*  ERROR         *  TRUE IFF AN ERROR OCCURRED
*
*****

```

FIGURE D.8

```

*****
*
*           C O M M O N   B L O C K   -   S O U R C E
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  ISOR1(I)      *  INDEX TO FIRST SOURCE CARD FOR SOURCE 1
*  NSOR(I)       *  NUMBER OF SOURCE CARDS FOR SOURCE 1
*  IFUNC(I)      *  GENERATING FUNCTION
*  IT1(I)        *  BEGINNING OF SOURCE AVAILABILITY
*  ITN(I)        *  END OF SOURCE AVAILABILITY
*  NPARM(I)      *  NUMBER OF PARAMETERS
*  IPARM1(I)     *  PARAMETER1
*  IPARM2(I)     *  PARAMETER2
*  IPARM3(I)     *  PARAMETER3
*  IPARM4(I)     *  PARAMETER4
*  IPARM5(I)     *  PARAMETER5
*  LBUCKET(I)    *  BUCKET SIZE
*  NSOUR         *  TOTAL NUMBER OF SOURCES
*  MXSOR         *  MAXIMUM NUMBER OF SOURCES
*  MXSORC        *  MAXIMUM NUMBER OF SOURCE CARDS
*  ERROR         *  TRUE IFF AN ERROR OCCURRED
*
*****

```

FIGURE D.9

```
*****
*
*      COMMON BLOCK - CONTROL (STEP3)
*
*****
*      *
* VARIABLE * DESCRIPTION
*      *
*****
* ICLOCK * SIMULATION TIME WHEN CLOCK WAS LAST CALLED
*      *
*****
```



FIGURE D.10

```

*****
*
*          COMMON BLOCK - RES
*
*****
*      *
* VARIABLE * DESCRIPTION
*      *
*****
*      *
* NRES      * NUMBER OF RESOURCES
* IBUCKET(I) * RESOURCE BUCKET SIZES
*      *
*****

```

FIGURE D.11

```

*****
*
*           COMMON BLOCK - R S U R C
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IAVAIL        *  POINTS TO FIRST AVAILABLE CELL
*  NAVAIL        *  NUMBER OF AVAILABLE CELLS
*  MAVAIL        *  MINIMUM OF CELLS TO BE RESERVED FOR FUTURE ADDS
*  ITIME(I)      *  TIME IN CELL I
*  IQUNT(I)      *  QUANTITY IN CELL I
*  LINK(I)       *  LINK IN CELL I
*  IFIRST(I)     *  POINTS TO BEGINNING OF LIST FOR RESOURCE I
*  ILAST(I)      *  POINTS TO END OF LIST FOR RESOURCE I
*  ITIME1(I)     *  EARLIEST TIME IN CORE FOR RESOURCE I
*  ITIMEH(I)     *  LATEST TIME IN CORE FOR RESOURCE I
*
*****

```



FIGURE D.12

```

*****
*
*           C O M M U N   B L O C K   -   S O R
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NSOR          *  NUMBER OF SOURCES
*  IBUCKET(I)    *  SOURCE BUCKET SIZES
*  ICU(I)        *  LENGTH OF TIME SOURCE IS AVAILABLE (CU)
*
*****

```

FIGURE D.13

```

*****
*
*          C O M M O N   B L O C K   -   S O U R S E
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  IAVAIL      *  POINTS TO FIRST AVAILABLE CELL
*  NAVAIL      *  NUMBER OF AVAILABLE CELLS
*  MAVAIL      *  MINIMUM OF CELLS TO BE RESERVED FOR FUTURE ADDS
*  ITIME(I)    *  TIME IN CELL I
*  IQUNT(I)    *  QUANTITY IN CELL I
*  LINK(I)     *  LINK IN CELL I
*  IFIRST(I)   *  POINTS TO BEGINNING OF LIST FOR SOURCE I
*  ILAST(I)    *  POINTS TO END OF LIST FOR SOURCE I
*  ITIME1(I)   *  EARLIEST TIME IN CORE FOR SOURCE I
*  ITIMEH(I)   *  LATEST TIME IN CORE FOR SOURCE I
*
*****

```

FIGURE D.14

```

*****
*
*      COMMON BLOCK - CBLK
*
*****
*
*  VARIABLE      *  DESCRIPTION
*
*****
*
*  NCOURS      *  NUMBER OF COURSES
*  IGRAD(1)    *  GRADUATION BLOCK FOR COURSE 1
*  ICTYPE(1)   *  TYPE OF COURSE 1
*  IPTYPE(1)   *  PERSONNEL TYPE FOR COURSE 1
*  IPRIOR(1)   *  PRIORITY OF COURSE 1
*  MXSIZE(1)   *  MAXIMUM SIZE OF COURSE 1
*  IPEROD(1)   *  PERIOD OF COURSE 1
*  IEGRAD(1)   *  EARLIEST GRADUATION DATE FOR COURSE 1
*
*****

```

FIGURE D.15

```

*****
*
*          C O M M O N   B L O C K   -   B L K S
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IBLOCK(I)    *  LOCATION OF FIRST PROC, TASK, RUB, AND RUDB BLOCK
*  NBLOCK(I)    *  NUMBER OF PROC, TASK, RUB, AND RUDB BLOCKS
*  LBLOCK(I)    *  LENGTH OF PROC, TASK, RUB, AND RUDB BLOCKS
*  IWORD(I)     *  POOL OF STORAGE CONTAINING ALL BLOCKS
*
*****

```

FIGURE D.16

```

*****
*
*          COMMON BLOCK - STACK
*
*****
*
*  VARIABLE  *  DESCRIPTION
*
*****
*
*  NSTACK    *  NUMBER OF ITEMS IN STACK
*  MSTACK    *  MAXIMUM NUMBER OF ITEMS A STACK HOLDS
*  ISTACK    *  STACK
*  JSTACK    *  STACK
*
*****

```



Section 2.3  
DESCRIPTIONS OF ROUTINES

```

C***** ABIN *****
C*
C*          SUBROUTINE ABIN
C*
C*  PURPOSE
C*    READ AIR BASE PARAMETERS AND INITIALIZES VARIABLES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    2 MAY 1975
C*
C*****

```

```

***** ARBASE *****
C*
C*          SUBROUTINE ARBASE
C*
C*  PURPOSE
C*    CONTROLS THE FLOW BETWEEN SUBROUTINES THAT PROCESS
C*    AIR BASE INFORMATION.
C*
C*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*    CORTIM - CALCULATES DURATION OF LONGEST TRACK IN EACH COURSE
C*    MAIN   - READS ALL AIR BASE PARAMETERS
C*    DISTAC - CALCULATES DISTRIBUTION OF AIR CRAFT
C*    DISTCW - CALCULATES DISTRIBUTION OF CREW MEMBERS
C*    MNCREW - DETERMINES MINIMUM CREW
C*    CCTS   - CALCULATES GRADUATION REQUIREMENTS
C*    PMT     - CALCULATES PMT COURSE DEMANDS
C*    RECOVR  - GENERATES SOURCE OF RECOVERABLE COPILOTS
C*    PRTCRS  - PRINTS REPORT CONTAINING STUDENTS/COURSE/YEAR
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    5 MAY 1975
C*
*****

```

```

***** BLKNAM *****
C*
C*          BLOCK DATA
C*
C*  PURPOSE
C*    INITIALIZED VARIABLE NEEDED WHEN INPUTTING NAMES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
*****

```

```

C***** BLKIN *****
C*
C*          SUBROUTINE BLKIN
C*
C*  PURPOSE
C*    READS THE BLOCKS NEEDED TO DEFINE COURSES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    24 APRIL 1975
C*
C*****

```



```

C***** BLOCK *****
C*
C* SUBROUTINE BLOCK
C*
C* PURPOSE
C* RETURNS THE CONTENTS OF A BLOCK
C*
C* CALLING SEQUENCE
C* CALL BLOCK(IADDR,IARRAY)
C*
C* DESCRIPTION OF PARAMETERS
C*
C* * EXPLICIT INPUT *
C* IADDR - POINTS AT BLOCK WHOS CONTENTS IS DESIRED.
C*
C* * EXPLICIT OUTPUT *
C* IARRAY - CONTENTS OF BLOCK ARE PLACED IN THIS ARRAY.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 24 APRIL 1975
C*
C*****

```

```

C***** CCTS *****
C*
C*                               SUBROUTINE CCTS
C*
C* PURPOSE
C*   DETERMINES DEMANDS DUE TO DELIVERY OF AIR CRAFT AND
C*   ATTRITION.
C*
C* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*   NAME - FINDS THE ALPHANUMERIC NAME OF AIR BASE
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   30 APRIL 1975
C*
C*****

```

```

C***** CLOCK *****
C*
C*                               SUBROUTINE CLOCK
C*
C* PURPOSE
C*   UPDATES CLOCK TIME AND UPDATES SOURCE AND RESOURCE TABLES
C*
C* CALLING SEQUENCE
C*   CALL CLOCK(ITIME)
C*
C* DESCRIPTION OF PARAMETERS
C*
C*       * EXPLICIT INPUT *
C*   ITIME - TIME TO BE ASSIGNED TO CLOCK
C*
C* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*   RDNAME - INPUTS NAME TABLES
C*   BLKIN  - INPUTS COURCES
C*   INTRES - INITIALIZES RESOURCE TABLES
C*   INTSOR - INITIALIZES SOURCE TABLES
C*   UPDRES - UPDATES RESOURCE TABLES
C*   UPDSOR - UPDATES SOURCE TABLES
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   28 APRIL 1975
C*****

```

```

C***** CURS IN *****
C*
C*          SUBROUTINE CORSIN
C*
C*  PURPOSE
C*    READS ALL BLOCKS NEEDED TO DEFINE COURSES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
C*****

```





```

C*****  CORTIM  *****
C*
C*          SUBROUTINE CORTIM
C*
C*  PURPOSE
C*    CALCULATES THE DURATION OF THE LONGEST TRACK IN EACH COURSE.
C*
C*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*    MAXTRK - CALCULATES THE DURATION OF THE LONGEST TRACK
C*             IN A PARTICULAR COURSE
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    6 MAY 1975
C*
C*****

```

```

***** DISTAC *****
C*
C*
C*          SUBROUTINE DISTAC
C*
C*  PURPOSE
C*    DETERMINE THE DISTRIBUTION OF AIR CRAFT FROM THE
C*    INITIAL INVENTORIES AND DELIVLRIES FOR A GIVEN AIR BASE.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    30 APRIL 1975
C*
*****

```

```

C***** DISTCW *****
C*
C*
C* SUBROUTINE DISTCW
C*
C* PURPOSE
C* DETERMINE THE INITIAL DISTRIBUTION OF CREWS FOR A GIVEN
C* AIR BASE FROM TIME HISTORY CARDS
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 1 MAY 1975
C*
C*****

```

```

***** FMTNAM *****
C*
C*
C* SUBROUTINE FMTNAM
C*
C* PURPOSE
C*   READS NAMES IN STEP2 AND WRITES OUT FOR STEP3.
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   22 APRIL 1975
C*
*****

```





```

C***** GENRES *****
C*
C* SUBROUTINE GENRES
C*
C* PURPOSE
C* APPLIES GENERATING FUNCTION TO RESOURCES.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 28 APRIL 1975
C*
C*****

```



```

C***** GETSOR *****
C*
C* SUBROUTINE GETSOR
C*
C* PURPOSE
C*   READS QUANTITY OF A GIVEN SOURCE FOR A GIVEN PERIOD.
C*
C* CALLING SEQUENCE
C*   CALL GETSOR(ISOR,IT1IN,IT2IN,IT1OUT,IT2OUT,IARRAY)
C*
C* DESCRIPTION OF PARAMETERS
C*
C*   * EXPLICIT INPUT *
C*   ISOR   - SOURCE NUMBER
C*   IT1IN  - BEGINNING OF TIME INTERVAL REQUESTED
C*   IT2IN  - END OF TIME INTERVAL REQUESTED
C*
C*   * EXPLICIT OUTPUT *
C*   IT1OUT - BEGINNING OF TIME INTERVAL RETURNED
C*   IT2OUT - END OF TIME INTERVAL RETURNED
C*   IARRAY - ARRAY OF QUANTITIES RETURNED
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   28 APRIL 1975
C*
C*****

```

```

C***** GRADBK *****
C*
C*          SUBROUTINE GRADBK
C*
C*  PURPOSE
C*    ASSIGNS PROC BLOCKS WITHOUT RIGHT POINTERS AS GRADUATION
C*    BLOCKS FOR EACH COURSE.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    24 APRIL 1975
C*****

```

```

C***** INTRES *****
C*
C* SUBROUTINE INTRES
C*
C* PURPOSE
C* INITIALIZE RESOURCE TABLES
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 28 APRIL 1975
C*
C*****

```



```

C***** INTSR *****
C*
C* SUBROUTINE INTSR
C*
C* PURPOSE
C* INITIALIZE SOURCE TABLES
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 28 APRIL 1975
C*
C*****

```

```

C***** MAXTRK *****
C
C      FUNCTION MAXTRK
C
C      PURPOSE
C      CALCULATES THE DURATION OF THE LONGEST TRACK IN A COURSE
C
C      CALLING SEQUENCE
C      MAXTRK(IPTR)
C
C      DESCRIPTION OF PARAMETERS
C
C      * EXPLICIT INPUT *
C      IPTR  - POINTS AT GRADUATION PROC BLOCK
C
C      AUTHOR/PROGRAMMER
C      JOHN R. MENIG
C      CALSPAN CORPORATION
C      24 APRIL 1975
C*****

```

```

C***** MNCREW *****
C*
C* SUBROUTINE MNCREW
C*
C* PURPOSE
C* DETERMINES MINIMUM CREW DISTRIBUTION TO BE MAINTAINED
C* FOR A GIVEN AIR BASE.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 30 APRIL 1975
C*
C*****

```

```

C***** NAME *****
C*
C* SUBROUTINE NAME
C*
C* PURPOSE
C* RETURN A NAME FOR CODE NUMBER.
C*
C* CALLING SEQUENCE
C* CALL NAME(IAPRV,NUMBER,INAME)
C*
C* DESCRIPTION OF PARAMETERS
C*
C* * EXPLICIT INPUT *
C* IAPRV - ALPHANUMERIC NAME OF THE TYPE OF NAME BEING LOOKED UP*
C* NUMBER - CODE NUMBER OF NAME BEING LOOKED UP
C*
C* * EXPLICIT OUTPUT *
C* INAME - ALPHANUMERIC NAME BEING RETURNED
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 22 APRIL 1975
C*
C*****

```

```

C***** NUMBER *****
C*
C*          SUBROUTINE NUMBER
C*
C*  PURPOSE
C*    RETURNS A CODE NUMBER FOR A NAME.
C*
C*  CALLING SEQUENCE
C*    CALL NUMBER(IAPRV,NUMB,NAME)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    IAPRV - ALPHANUMERIC NAME OF THE TYPE OF CODE BEING LOOKED UP*
C*    NAME  - ALPHANUMERIC NAME BEING LOOKED UP
C*
C*          * EXPLICIT OUTPUT *
C*    NUMB  - CODE NUMBER RETURNED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
C*****

```



```
C***** PMT *****
C*
C*          SUBROUTINE PMT
C*
C*  PURPOSE
C*    CREATES PMT DEMANDS AND DETERMINES WHETHER PMT IS FEASIBLE
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    2 MAY 1975
C*
C*****
```

```

C***** PRCOUR *****
C*
C* SUBROUTINE PRCOUR
C*
C* PURPOSE
C* CONTROLS THE FLOW BETWEEN PROGRAMS THAT INPUT, REFORMAT,
C* EXAMINE, AND OUTPUT BLOCKS THAT DEFINE COURSES.
C*
C* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C* CORGIN - READS ALL BLOCKS NEEDED TO DEFINE COURSES.
C* PRPROC - CHANGES PROC AND TASK BLOCK NUMBERS INTO POINTERS IN
C* PROC BLOCKS.
C* PRTASK - CHANGES RUB NUMBERS INTO POINTERS IN TASK BLOCKS.
C* PRRUB - CHANGES RUDB NUMBERS INTO POINTERS IN RUB BLOCKS.
C* PRRUDB - CHANGES RUB AND RUDB NUMBERS INTO POINTER IN RUDB
C* BLOCKS
C* GRADBK - FINDS GRADUATION BLOCKS FOR EACH COURSE
C* TSSYN - TEST THAT THE SYNCHRONIZED PROC BLOCKS FOR A
C* CIRCULAR LIST.
C* PROCLP - TEST THAT PROC BLOCKS TERMINATE
C* RUDGLP - TEST RUB AND RUDB COMBINATIONS
C* SORTLK - IN PROC BLOCKS SORTS LEFT LINKS BY PRIORITY
C* CORSOT - WRITES ALL BLOCKS NEEDED TO DEFINE COURSE IN STEP3.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 24 APRIL 1975
C*
C*****

```

```

C***** PROCLP *****
C*
C*          SUBROUTINE PROCLP
C*
C*  PURPOSE
C*    TEST THAT EACH TRACK IN A COURSE IS SHORTER THAN A
C*    PREDETERMINED MAXIMUM LENGTH.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    24 APRIL 1975
C*
C*****

```

```

C***** PRPROC *****
C*
C* SUBROUTINE PRPROC
C*
C* PURPOSE
C* SUBSTITUTES POINTERS FOR PROC BLOCK NUMBERS AND TASK NUMBERS;
C* SUPPLIES RIGHT PROC BLOCK POINTERS IN PROC BLOCKS.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 23 APRIL 1975
C*
C*****

```

```

C***** PRRUB *****
C*
C*          SUBROUTINE PRRUB
C*
C*  PURPOSE
C*    SUBSTITUTES POINTERS FOR RUDB NUMBERS IN RUB BLOCKS.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    23 APRIL 1975
C*
C*****

```



```

C***** PRRUDB *****
C*
C* SUBROUTINE PRRUDB
C*
C* PURPOSE
C* SUBSTITUTES POINTERS FOR RUB AND RUDB NUMBERS IN RUDB BLOCKS.*
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 23 APRIL 1975
C*
C*****

```

```

C***** PRTASK *****
C*
C*          SUBROUTINE PRTASK
C*
C*  PURPOSE
C*    SUBSTITUTES POINTERS FOR RUB NUMBERS IN TASK BLOCKS.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    23 APRIL 1975
C*
C*****

```

```

C***** PRTCKS *****
C*
C*          SUBROUTINE PRTCKS
C*
C*  PURPOSE
C*    PRINTS THE NUMBER OF STUDENTS/COURSE/YEAR
C*
C*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*    NAME - FINDS THE ALPHANUMERIC NAME OF COURSES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 MAY 1975
C*****

```

```

C***** PUTRES *****
C*
C*          SUBROUTINE PUTRES
C*
C*  PURPOSE
C*    WRITES QUANTITY OF A GIVEN RESOURCE FOR A GIVEN PERIOD
C*
C*  CALLING SEQUENCE
C*    CALL PUTRES(IRES,IT1,IT2,IARRAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    IRES  - RESOURCE NUMBER
C*    IT1   - BEGINNING OF INTERVAL
C*    IT2   - END OF INTERVAL
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    29 APRIL 1975
C*
C*****

```

```

C***** PUTSOR *****
C*
C*          SUBROUTINE PUTSOR
C*
C*  PURPOSE
C*    WRITES QUANTITY OF A GIVEN SOURCE FOR A GIVEN PERIOD.
C*
C*  CALLING SEQUENCE
C*    CALL PUTSOR(ISOR,IT1,IT2,IARRAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    ISOR  - SOURCE NUMBER
C*    IT1   - BEGINNING OF INTERVAL
C*    IT2   - END OF INTERVAL
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    29 APRIL 1975
C*
C*****

```



```

C***** RDNAME *****
C*
C*          SUBROUTINE RDNAME
C*
C*  PURPOSE
C*    READS NAMES IN STEP3
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
C*****

```

```

C***** RECOVER *****
C*
C* SUBROUTINE RECOVER
C*
C* PURPOSE
C* CREATE COPILOT SOURCES
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 2 MAY 1975
C*
C*****

```

```

C***** RESR IN *****
C*
C* SUBROUTINE RESR IN
C*
C* PURPOSE
C* READ RESOURCES, TEST RESOURCES FOR ERRORS, AND DETERMINE
C* BUCKET SIZES.
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 25 APRIL 1975
C*
C*****

```

```

C***** RESSOR *****
C*
C* SUBROUTINE RESSOR
C*
C* PURPOSE
C* CONTROLS FLOW BETWEEN ROUTINES THAT PROCESS RESOURCES AND
C* SOURCES IN STEP2.
C*
C* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C* GENRES - GENERATES RESOURCES FROM RESOURCE CARDS
C* GENSOR - GENERATES SOURCES FROM SOURCE CARDS
C* RESRIN - READS RESOURCE CARDS
C* SOURIN - READS SOURCE CARDS
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 28 APRIL 1975
C*
C*****

```

```

C***** RUDBLP *****
C*
C* SUBROUTINE RUDBLP
C*
C* PURPOSE
C* TEST THAT THE DEPTH OF RUDB AND RUB COMBINATIONS ARE LESS
C* THAN A PREDETERMINED MAXIMUM
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 25 APRIL 1975
C*
C*****

```



```

C***** SORTLK *****
C*
C*          SUBROUTINE SORTLK
C*
C*  PURPOSE
C*    SORTS LEFT LINKS BY ASCENDING PRIORTIES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    15 MAY 1975
C*
C*****

```

```

C***** SOURIN *****
C*
C*          SUBROUTINE SOURIN
C*
C*  PURPOSE
C*    READS SOURCES, TESTS SOURCES FOR ERRORS, AND DETERMINES
C*    BUCKET SIZES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    25 APRIL 1975
C*
C*****

```

```

C***** STEP2 *****
C*
C*          MAIN PROGRAM STEP2
C*
C*  PURPOSE
C*    CALLS ROUTINES TO INPUT NAMES, RECONSTRUCT COURSE BLOCKS,
C*    CALCULATE AIR BASE DEMANDS, GENERATE RESOURCES, AND
C*    GENERATE SOURCES.
C*
C*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*    FMTNAM - INPUTS NAMES
C*    PROCUR - RECONSTRUCTS COURSES
C*    ARBASE - CALCULATES AIR BASE DEMANDS
C*    RESSOR - GENERATES RESOURCES AND SOURCES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    6 MAY 1975
C*
C*****

```

```

C***** TSSYN *****
C*
C*          SUBROUTINE TSSYN
C*
C*  PURPOSE
C*    TESTS THAT PROC BLOCKS THAT ARE SYNCHRONIZED FORM A CIRCULAR
C*    LIST CONTAINING MORE THAN ONE BLOCK BUT LESS THAN A
C*    PREDETERMINED MAXIMUM NUMBER OF BLOCKS.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    23 APRIL 1975
C*
C*****

```

```

C***** UPDRES *****
C*
C* SUBROUTINE UPDRES
C*
C* PURPOSE
C*   UPDATES RESOURCE LISTS WHEN CLOCK TIME HAS CHANGED
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   28 APRIL 1975
C*
C*****

```



```

C***** UPDSOR *****
C*
C*          SUBROUTINE UPDSOR
C*
C*  PURPOSE
C*    UPDATES SOURCE LISTS WHEN CLOCK TIME HAS CHANGED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*
C*****

```

Section 2.4

CROSS REFERENCE TABLES FOR ROUTINES AND VARIABLES  
USED IN COMMONS

# CROSS REFERENCE USAGE CODES

## A ARGUMENT

THE SYMBOL IS A VARIABLE OR FUNCTION NAME WHICH APPEARS IN AN ARGUMENT LIST OF A CALL, SUBROUTINE, FUNCTION, OR ENTRY STATEMENT.

## D DATA INITIALIZATION

THE SYMBOL IS A VARIABLE WHICH IS INITIALIZED IN A DATA OR TYPE SPECIFICATION STATEMENT SUCH AS A COMPLEX SPECIFICATION STATEMENT.

## F FEICH A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE MOST RECENTLY ASSIGNED VALUE IS ACCESSED BUT NOT CHANGED.
2. FUNCTION NAME OR ARGUMENT OF A FUNCTION WHICH APPLIES ON THE RIGHT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT OR APPEARS IN AN IF STATEMENT TEST.
3. DUMMY ARGUMENT IN A STATEMENT FUNCTION DEFINITION.

## S STORE A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE VALUE IS REPLACED BY ANOTHER VALUE.
2. FUNCTION NAME WHICH APPEARS ON THE LEFT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT.
3. NAME OF A STATEMENT FUNCTION IN THE DEFINITION OF THAT FUNCTION.

## C COMMON

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A COMMON STATEMENT OR IS THE NAME OF A LABELED COMMON BLOCK.

## E EQUIVALENCE

THE SYMBOL IS A VARIABLE WHICH APPEARS IN AN EQUIVALENCE STATEMENT.

## T TYPE SPECIFICATION

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A:

1. TYPE SPECIFICATION STATEMENT AND IS NOT INITIALIZED IN THAT STATEMENT.
2. DIMENSION OR EXTERNAL STATEMENT.

## N ENTRY POINT

THE SYMBOL IS AN ENTRY POINT DEFINED BY AN ENTRY STATEMENT IN A SUBROUTINE OR FUNCTION.

## X EXTERNAL REFERENCE

THE SYMBOL IS A SUBROUTINE OR ENTRY NAME WHICH APPEARS IN A CALL STATEMENT.

CRUSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGL SUMMARY

MAIN	PRCCUR	CURSN	PRPRCC	PRTASK	PRRJD	PRRUDb	GRAJDB	ISSYN	PRUCLP	KUDBLP	SORTLN
ALERTR	R										
C	R										
CH	R										
CONTRL	CB										
CPATTR	R										
CRECY	R										
CREW	R										
CREWR	R										
D	R										
DM	R										
DPATTR	R										
DUMMY	CB										
ERROR	L										
HRBUCK	R										
HRCRBK	R										
HRSCU	R										
IAB	I										
IABCN	I										
IABCI	I										
IABON	I										
IABDI	I										
IABEN	I										
IABEI	I										
IABPCN	I										
IABPCI	I										
IABPN	I										
IABPI	I										

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

		CURSUT	ABASE	ADIN	CCTS	PAT	RECOVR	DISTAC	DISTCH	MNCKEW	CURTIM	MAXTAK	PRICKS
ALERTR	R			SC		FC							
C	R			C	FSC	FC			SC				
CH	R			C		FSC							
CONTAL	CB			C									
CPATTR	R			FSC		C							
CRECY	R			SC	FSC								
CREW	R			C	FC	FC				SC			
CREWR	R			SC		C				FC			
D	R			C	FSC	FC			SC				
DH	R			C	C	FSC							
DPATTR	R			FSC	FC	C							
DUMMY	CB			C		C							
ERROR	L			SC	SC	SC				SC			
HRBUCK	R			SC		FC							
HRCRBK	R			SC		FC							
HRSCU	R			FSC		FC							
IAB	I			FSC	AFC	FC				FC			
IABCN	I			FSC		C							
IABCL	I			SC	FC	C							
IABDN	I			FSC		C							
IABDI	I			SC		C							
IAPEN	I			FSC		FC							
IABE1	I			SC		FC							
IABPCN	I			FSC		FC							
IABPC1	I			SC		FC							
IABPN	I			FSC		FC							
IABP1	I			SC		FC							



CROSS REFERENCE SUMMARY \*\*\*\*\* STEP \*\*\*\*\*

SYMBOL	TYPE	FMNAM	BLOCK	NAME	NUMBER	GENRES	GENSUR	RESRIN	RESSUR	SOURIN
ALERTK	R									
C	R									
CH	R									
CONTRL	CB									
CPATTR	K									
CRECY	K									
CREW	R									
CREWR	R									
D	R									
DH	R									
DPATTR	R									
DUMMY	CB									
ERROR	L									
HRBUCK	R									
HRCRBK	R									
HNSCU	R									
IAB	I									
IABCN	I									
IABCI	I									
IABDN	I									
IABDI	I									
IABEN	I									
IABEL	I									
IABPCN	I									
IABPCI	I									
IABPN	I									
IABPI	I									

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

SYMBOL	TYPE	MAIN	PRCOUR	CURSIN	PRPRGC	PRTASK	PRRUB	PRRDOB	GRADBK	ISSYN	PKUCLP	RUEBLP	SORTLK
IAC	I												
IADAIL	I												
IBLOCK	I												
IBUCKT	I												
ICATTD	I												
ICBUCK	I												
ICOUPC	I												
ICOURC	I												
IGRECT	I												
ICTYPE	I												
ICUVR	I												
IDATEC	I												
IDATED	I												
IDATEE	I												
IDATEP	I												
IDATPC	I												
IDATTU	I												
IEGRAD	I												
IFIRST	I												
IFUNC	I												
IGRAD	I												
ILAST	I												
INVAC	I												
INVC	I												
INVD	I												
INVO	I												
INVP	I												

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	CORSOT	ARBASE	ALIN	CCIS	PNI	RECOVR	DISTAC	DISTCW	MNCRLW	CORTIM	MAXTRK	PRICKS
IAC		C	L	FC		C	FSC	C	FC			
IABAIL	C									C	C	
IBLOCK	FC									C	C	
IBUCKT		C	FSC	FC	FC	FC	C	C	C			
ICATTU		C	FSC	FC	C	C	C	C	C			
ICBUCK			SC									
ICJUPC		C	FSC	C	FC	C	C	C	C			
ICOURC		C	FSC	FC	C	C	C	C	C			
IGRECT			FSC									
ICTYPE	FC		FSC									
ICUVR		C	FSC	FC	FC	FC	C	C	C			
IDATEC		C	SC	FC	C	C	C	C	C			
IDATED		C	SC	C	C	C	FC	C	C			
IDATEE		C	SC	C	FC	C	C	C	FC			
IDATEP		C	SC	C	FC	C	C	C	C			
IDATPC		C	SC	C	FC	C	C	C	C			
IDATTU		C	FSC	FC	C	C	C	C	C			
IEGRAD	FC									C	C	
IFIRST					FS		FS		FS			
IFUNC												
IGRAD												
ILAST	FC				FS	FS	FS		FS	FC	C	
INVAL		C	SC	C	C	C	FC	C	C			
INVC		C	SC	FC	C	C	C	C	C			
INVD		C	SC	FC	C	C	C	C	C			
INVO		C	SC	C	C	C	C	C	C			
INVP		C	SC	FC	C	C	C	C	C			

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL	TYPE	FMNAM	*BLOCK	NAME	NUMBER	GENRES	GENSUR	RESKIN	RESSOR	SOURIN
IAC	I									
IADAIL	I									
IBLOCK	I									
IBUCKT	I									
ICATTD	I									
ICBUCK	I									FC
ICOUPC	I									
ICOURC	I									
ICRECT	I									FC
ICTYPE	I									
ICUYR	I									
IDATEC	I									
IDATED	I									
IDATEE	I									
IDATEP	I									
IDATPC	I									
IDATTD	I									
IEGRAD	I									
IFIRST	I	FSC	C	FC	FC					
IFUNC	I					C	C	FSC		FSC
IGRAD	I									
ILAST	I									
INVAC	I									
INVL	I									
INVD	I									
INVD	I									
INVD	I									
INVP	I									

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	MAIN	PCOUR	CURSIN	PRPRUC	PRTASK	PRRJE	PREUDB	GRADBN	ISSN	PRCLUP	RUGDLP	SERILK
IOATTD	I												
IPARM1	I												
IPARM2	I												
IPARM3	I												
IPARM4	I												
IPARM5	I												
IPATTD	I												
IPERC	I												
IPEROD	I		C	SC	C	C	C	C	C	C	C	C	C
IPERTP	I												
IPMT	I												
IPRIOR	I		C	SC	C	C	C	C	C	C	C	C	C
IPTYPE	I		C	SC	C	C	C	C	C	C	C	C	C
IQANTD	I												
IRESI	I												
ISORI	I												
ISTACK	I										FSC	FSC	
ITIME	I												
ITIMEM	I												
ITIMEL	I												
ITL	I												
ITN	I												
ITYPE	I												
ITL	I												
IUNII	I			IFS									
IWORD	I		C	SC	FSC	FSC	FSC	FSC	FSC	FSC	FSC	FSC	FSC
JPMT	I												



CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

		CURSDT	ARBASE	ABIN	CCIS	PMT	RECOVR	DISTAC	DISTCM	MNUKEM	CORTIM	MAXTRK	PRICRS
IUATTO	I												
IPARM1	I												
IPARM2	I												
IPARM3	I												
IPARM4	I												
IPARM5	I												
IPATTD	I												
IPERC	I												
IPEROD	I												
IPERTP	I												
IPMT	I												
IPRIOR	I												
IPTYPE	I												
IQANTD	I												
IRES1	I												
ISORI	I												
ISTACK	I												
ITIME	I												
ITIMEH	I												
ITIMEL	I												
ITL	I												
ITN	I												
ITYPE	I												
ITI	I												
IUNIT	I												
IWORD	I												
JPMT	I												

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

PMINAM	*BLOCK	NAME	NUMBER	GENES	GENSUK	RESRIN	MESSUR	SOURIN
IGATTD								
IPARM1				FC	FC	SC		SC
IPARM2				FC	FC	FSC		FSC
IPARM3				C	C	SC		SC
IPARM4				C	C	SC		SC
IPARM5				C	C	SC		SC
IPATTD								
IPERC								
IPERQD								
IPERTP								
IPMT								
IPRIOR								
IPTYPE								
IQANTD								
IRES1				FC		SC		
ISUR1					FC			SC
ISTACK								
ITIME								
ITIMEH				FS	FS			
ITIMEL				FS	FS			
ITL								
ITN				FC	FC	SC		SC
ITYPE								
ITL								
IUNIT				FC	FC	SC		SC
IWORD				LF	UF	UF		UF
JPMT								

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		MAIN	PRACOUR	CURSIN	PRPRDC	PRTASK	PRRUE	PRRUDB	GRADBK	ISSYN	PRACCLP	RUECLP	SURTILK
JSTACK	I										FSC		
JUJIT	I												
LBKIN	I		C	FSC	C	C	C	C	C	C	C	C	C
LBKOUT	I		C	FSC	CE	CE	CE	CE	CE	C	CE	CE	CE
LEUCKT	I												
MAB	I												
MABC	I												
MABD	I												
MABE	I												
MABH	I												
MABP	I												
MABPC	I												
MAXLEN	C6												
MAXNUM	I												
MAXTIM	I												
MAXTRK	I												
MBUCT	I												
MSTACK	I									FSC	FSC		
MBUCK	I												
MACOUR	I		C	FSC	C	C	C	C	C	C	C	C	C
MXRES	I												
MXRESC	I												
MXSIZE	I		C	SC	C	C	C	C	C	C	C	C	C
MXSUR	I												
MXSORC	I												
MAXIME	I												
NAB	I												

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*  
 SYMBOL TYPE \*\*\*\*\* USAGE SUMMARY \*\*\*\*\*

	CORSUT	AREASE	ABIN	CCIS	PMT	RECOVR	DISTAC	DISTCH	MINCKW	LCRTIM	MAXTRK	PRTCRS
JSTACK	I											
JUNIT	I											
LBKIN	I											
LBKOUT	I											
LBCKT	I											
MAB	I											
MABC	I											
MABD	I											
MABE	I											
MABH	I											
MABP	I											
MABPC	I											
MAXLEN	CB											
MAXNUM	I											
MAXTIM	I											
MAXTRK	I											
MBUCT	I											
MSTACK	I											
MXBUCK	I											
MXCOUR	I											
MXRES	I											
MXRESC	I											
MXSIZE	I											
MXSOR	I											
MXSORC	I											
MXTIML	I											
NAB	I											

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL	TYPE	FMINAM	*BLOCK	NAME	NUMBER	GENRES	GENSUR	RESRIN	RESSCR	SCUPIN
JSTACK	I									
JUNIT	I	F C	D C	C	C			DF		DF
LBKIN	I									
LBKOUT	I									
LBUCKT	I					F C	F C	FSC		FSC
MAB	I									
MABC	I									
MABD	I									
MABE	I									
MABH	I									
MABP	I									
MABPC	I									
MAXLEN	CB									
MAXNUM	I	F C	D C	C	C					
MAXTIM	I									
MAXTRK	I									
MBUCKT	I									
MSTACK	I									
MXBUCK	I									
MXCOUR	I									
MXRES	I					C		FSC		
MXRESC	I					C		FSC		
MXSIZE	I									
MXSOR	I						C			FSC
MXSURC	I						C			FSC
MXTIME	I					F C	F C			
NAB	I									

USAGE SUMMARY



CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	PKCOUR	CORSIN	PKPRUC	PRTASK	PRRUB	PRRUDG	GRADBK	TSSYN	PRCCLP	RUDBLP	SURTLK
NABC	I											
NABD	I											
NABE	I											
NABH	I											
NABP	I											
NABPC	I											
NAM	CB											
NAME	I											
NAMES	I											
NBLOCK	I											
NCORS	I											
NCOURS	I											
NOPRNT	L											
NPARH	I											
NRES	I											
NRESR	I											
NSUR	I											
NSOUR	I											
NSTACK	I											
NTYPE	I											
NUM	I											
NUMBER	I											
NWORDS	I											
NYEARS	I											
U	R											
UH	R											
UPATTH	R											

GROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

		CURSOR	AKBASE	ALIN	CCTS	PMT	RECOVER	DISTAC	DISTCH	MNCREW	LOKIM	MAXINK	PRICKS
NABC	I		C	FSC	C	C	C	C	C	C			C
NASU	I		C	FSC	C	C	C	C	C	C			C
NABE	I		C	FSC	C	C	C	C	C	C			C
NABH	I		C	FSC	C	C	C	C	C	C			C
NABP	I		C	FSC	C	C	C	C	C	C			C
NABPL	I		C	FSC	C	C	C	C	C	C			C
NAM	Co												
NAME	I												
NAMES	I												
NBLOCK	I	FC									C		
NCORS	I		C	FSC	C	C	C	C	C	C			FC
NCUORS	I	FC				FS					FC		FC
NOPRNT	L		C	FSC	FC	CT	FC	CT	CT	CT			FC
NPARM	I												
NRES	I												
NRESK	I												
NSOR	I												
NSOUR	I											FSC	
NSTACK	I												
NTYPE	I												
NUM	I												
NUMBER	I												
NWORS	I												
NYEARS	I		C	FSC	C	C	FC	C	C	C	C		FC
U	R		C	C	FSC	FC	C	C	SC	C			C
OH	R		C	C	C	FSC	C	C	C	C			C
OPATTR	R		C	FSC	FC	C	C	C	C	C			C

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

JYMBUL	TYPE	FMINAM	*BLOCK	NAME	NUMBER	GENRES	GENSUR	RESKIN	RESSUR	SCUFIN
NAB C	I									
NAB U	I									
NAB E	I									
NAB H	I									
NAB P	I									
NAB PC	I									
NAM	CB	C	C	C	C					
NAME	I				A F T					
NAMES	I			F C	F C					
NBLOCK	I									
NCORS	I									
NGOURS	I									
NOPRNT	L									
NPARM	I					C	C	FSC		FSC
NRES	I					F C		FSC		
NRESR	I					F C		FSC		
NSOR	I						F C			FSC
NSOUR	I						F C			FSC
NSTACK	I									
NTYPE	I	F C	D C	F C	F C					
NUM	I	FSC	C	F C	F C					
NUMBER	I			A F						
NWORDS	I									
NYEARS	I									
U	R									
JH	R									
OPATTR	R									

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	PROCUR	CURSIN	PKPRCC	PRTASK	PRRUE	PRRUDb	GRAEBK	ISSYN	PRCLLP	ROGBLP	SORTLK
P												
PCC												
PCPC												
PCRECY												
PH												
PPATTR												
RECVRY												
RESUMS												
SOURCE												
STACK												
STUDS												
WKSOKT												

CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	CURSOT	ANBASE	ABIN	CCTS	PMT	RECOVR	DISTAL	DISTCH	MNCKEM	LOKTIM	MAXTRA	PRICRS
P		C	C	FSC	F	C	C	SC	C			C
PCC		C	SC	F	C	C	C	C	C			C
PCPL		C	SC	C	F	C	C	C	C			C
PCRECY		C	FSC	C	C	F	C	C	C			C
PH		C	C	C	FSC	C	C	C	C			C
PPATTR		C	FSC	F	C	C	C	C	C			C
RECVMY			C									
RESURS												
SOURCE												
STACK											C	
STUDS		C	SC	FSC	FSC	C	C	C	C			F
WKSBLT		C	FSC	C	C	C	C	C	C			C



CROSS REFERENCE SUMMARY \*\*\*\*\* STEP \*\*\*\*\*

SYMBOL	TYPE	FMTNAM	#BLOCK	NAME	NUMBER	GENRES	GENSUR	RESRIN	KESSLR	SUOKIN
P	R									
PCC	R									
PCPC	R									
PCRECY	R									
PH	R									
PPATTR	R									
RECVR	Cb									
RESURS	Cb									
SOURCE	Co									
STACK	Cb									
STUDS	R									
WKSBLT	R									

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	MAIN	PRCUUR	CURSIN	PRPRCL	PRTASK	PRRUB	PRRUBB	GRABK	ISSYN	PRCLP	RUELP	SURTILK
ABIN												
ARBASE	X											
LCTS												
CURSIN		X										
CORSOT		X										
CURTIM												
DISTAL												
DISTCH												
FMTNAM	X											
GENRES												
GENSOR												
GRABK		X										
HNCREW												
NAME												
PMT												
PRCUUR	X											
PROCLP		X										
PRPRUC		X										
PRRUB		X										
PRRUCB		X										
PRTASK		X										
PRTCKS												
RECOVR												
NESSOK	X											
RUEBLP		X										
SURTILK		X										
SOURIN												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

ROUTINE OR ENTRY	CORSUT	ARBASE	AEIN	CCTS	PMT	RECUVR	DISTAC	DISTCH	MNCREW	CORTIN	MAXTRK	PRTRKS
ABIN		X										
ARBASE												
CCTS		X										
CORSIN												
CORSOT												
CORTIM		X										
DISTAL		X										
DISTCH		X										
FMTNAM												
GENRES												
GENSOR												
GRADBK												
MNCREW		X										
NAME				X								X
PMT		X										
PRCOUR												
PROCLP												
PRPRDC												
PRRUB												
PRAUDB												
PRTASK												
PRTRCS		X										
RECOVM		X										
KESSON												
RUDBLP												
SCATLK												
SCURIN												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

ROUTINE OR ENTRY	FMTNAM	BLUOK	NAME	NUMBLK	GENRES	GENSOR	RESKIN	LESSOR	SLURIN
ABIN									
ARBASL									
CCTS									
CORSIN									
CORSOT									
CORTIM									
DISTAC									
DISTCH									
FMTNAM								X	
GENRES								X	
GENSOR									
GRADBK									
MNCREW									
NAME									
PMT									
PRCOUR									
PRGCLP									
PRPROC									
PRRUB									
PRRUDB									
PRTASK									
PRTCRS									
RECOVR									
RESSJR									
RUDBLP									
SORTLK									X
SOURIN									

SUBROUTINE CROSS REFERENCE SUMMARY ..... STEP2 .....

ROUTINE  
OR ENTRY

USAGE SUMMARY

MAIN	PKCUJR	CURIN	PKPROC	PKTASK	PKRUB	PKRUBB	GRADEN	TSJVN	PRCLP	KULCLP	SORTLK
1	1	1	1	1	1	1	1	1	1	1	1
TSYN	1	1	1	1	1	1	1	1	1	1	1



SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

ROUTINE  
OK ENTRY

USAGE SUMMARY

	CORSOT	AKDASE	AEIN	UCTS	PMT	RECUVR	UJSTAL	UJSTCL	MNCLNEM	CL-LJIM	MAXTRN	PRJCLAS
TSSYN												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* STEP2 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

FMTNAM	4BLOCK	NAME	NUMBER	GENES	GENSUK	RESRIN	RESSUR	SUKRIN

TSSYN

## Section 3.0

### TRAM PHASE 3

#### 3.1 Introduction

The purpose of this section is to supplement the data in Technical Memorandum SAT-5, TRAM User's Guide with respect to Phase 3 of TRAM. This Programmer's Guide consists of a description of the data management system used in Phase 3, a listing of the input and output data sets, subroutine description and flow diagrams, cross reference tables and block descriptions.

In addition to the description of the data management system, the reader should note the capability for varying the dimensions of Phase 3 as described in Section 7.

##### 3.1.1 Data Management

Because of the dynamic data flow in the Phase 3 TRAM Program, the standard FORTRAN array and indexing structures are inadequate in terms of core utilization and computational efficiency.

Most of the information used by the program is grouped into blocks of data that are organized using singly linked lists. The formats of the different blocks used are presented in Section 3.7. This method makes it possible to add and delete blocks to the lists without a need for periodic reorganization.

The procblocks, task blocks, resource utilization blocks (RUBs) and resource utilization description blocks (RUDBs) share a common pool of storage in common BLKS and are accessed directly by their addresses. Subroutine BLOCK is used to copy any of these blocks into local storage.

#### 3.2 Description Of Inputs

The inputs consist of a limited number of cards described in SAT-5, TRAM User's Guide and the following data sets:

### Training Demand Records

Training demand records (Figure 3.1) are written out by the Phase 2 TRAM program on either tape or disk. They are 6 words long and written without using a format statement.

Before use in Phase 3 of TRAM, the training demand records are sorted on time in decreasing order.

### Resource Inventories

The resource inventory records (Figure 3.2) are written out by the Phase 2 TRAM program on either disk or tape. They are 3 words long and written without using a format statement.

The resource records are sorted in decreasing order by time.

### Trainee Inventories

The source records (Figure 3.3) describe the trainee inventories. These records are written by the Phase 2 TRAM program on either disk or tape. They are 3 words long and are written without using a format statement.

The source records are sorted in decreasing order by time.

### Description Of Training Program

The Training Program (also referred to as courses) is described by means of Procblocs, Task Blocks, Resource Utilization Blocks and Resource Utilization Description Blocks. The detailed formats of these data blocks are given in Section 3.7.

These blocks are read into core from FORTRAN Unit 20 when the CLOCK subroutine is invoked for the first time. The addresses of the first procbloc for each course (the Graduation Block) are stored in array IADPB1 in common CBLK. Each procbloc points to the procbloc(s) lying to the left and right of it and to the tasks associated with it. Task blocks point to RUBs and RUBs

Figure 3.1

```

*****
*
*   T R A I N I N G   D E M A N D   R E C O R D
*
*****
*
*   WORD      *   D E S C R I P T I O N
*
*****
*
*   1   *TIME
*   2   *QUANTITY (FLOATING POINT NUMBER OF TRAINEES).
*   3   * TYPE OF PERSONNEL. 1- PILOTS
*           2- COPILOTS
*           3- OSOS
*           4- DSOS
*   4   *COURSE NUMBER
*   5   *DEMAND NUMBER = AIR BASE NUMBER * 1000 + BUCKET NO.
*   6   *DEMAND TYPE.  1-CCTS BECAUSE OF DELIVERIES.
*           2-CCTS BECAUSE OF ATTRITION.
*           3-PMP
*
*****

```



Figure 3.2

```
*****
*
*   RESOURCE   RECORD
*
*****
*
* WORD      * DESCRIPTION
*
*****
*
*   1      * TIME
*   2      * RESOURCE NUMBER
*   3      * QUANTITY
*
*****
```

Figure 3.3

```
*****
*
*   S O U R C E   R E C O R D
*
*****
*           *
*  WORD    *  D E S C R I P T I O N
*           *
*****
*           *
*    1     *  T I M E
*    2     *  S O U R C E   N U M B E R
*    3     *  Q U A N T I T Y
*           *
*****
```

in turn point to the RUDBs. This linked structure permits quick access (using subroutine BLOCK) to information required for performing the different functions of the program (i.e., Class Transfer Tasks, Resource Utilization Tasks, etc.)

Class blocks, stored in common CLASSB, are created for each new class of students entering a course at the graduation block and for every time that an existing class is split among different tracks. Class blocks are deleted from the list whenever a procblock without a left branch is executed. Subroutine NEWCLS creates class blocks and subroutine REMCLS deletes them.

Predetermined transfer blocks, stored in common PTBC, are created by subroutine FRMPBTB when a source allocation task (SCATSA) is executed. The pointer to the first PTB is placed in the class block. After a PTB is used to control a class transfer at a node, it is deleted and the space it used is released by subroutine REMPTB. The pointer in the class block is updated to point to the next PTB.

Look-up and updates of resource and source inventories are done by using subroutines GETRES, PUTRES, GETSOR and PUTSOR. Resource and source inventories are stored on tape or disk. When subroutine clock is called for the first time, the buffers allocated to the inventories are filled with data starting at the simulation clock time and extending as far back as space permits. Every time that the subroutine clock is called, inventory records for times greater than the simulation clock time are written out on tape or disk, and the core thus made available is used to read in resource and source inventories for an earlier time.

### 3.3 Description Of Outputs

The outputs of the Phase 3 TRAM program consist of:

1. Echo of inputs.
2. Resource inventories remaining after training demands have been satisfied.
3. Trainee (Source) inventories remaining after training demands have been satisfied.
4. Lag records.
5. Source allocation records.
6. Warning and error and normal end messages.

Items 2-5 are described below. Items 1 and 6 are described in detail in SAT-5.

#### Resource Inventories

The output resource inventory records are identical in form to the input resource inventory records (Figure 3.2).

The input inventory minus the output inventory for any given time interval is the amount of the resource consumed during that time to satisfy the training requirements.

#### Source Inventories

The output source inventory records are identical in form to the input source inventory records (Figure 3.3).

The input inventory minus the output inventory for any time interval is the number of trainees from that particular source actually assigned to the training program during that time interval.

### Lag Records

The lag records (Figure 3.4) are written out on tape or disk by the Phase 3 TRAM program whenever a class has to be lagged.

Note - Processing in TRAM 3 is done in reverse time order (i.e. last PROCBLOC of a course is done first, first PROCBLOC is done last.) Thus when a class is lagged, the net effect is to force something to occur at an earlier date.



Figure 3.4

```

*****
*
*           L A G   R E C O R D
*
*****
*
*  WORD      *  D E S C R I P T I O N
*
*****
*
*  1      *  1
*  2      *  CLASS ADRESS
*  3      *  CURRENT CLASS TIME (TIME AT WHICH LAG STARTS).
*  4      *  COURSE NUMBER
*  5      *  PROCBLOCK NUMBER
*  6      *  TASK NUMBER
*  7      *  0
*  8      *  0
*  9      *  DURATION OF CURRENT PROCBLOCK.
*  10     *  LAG DURATION
*  11     *  LAG REASON.      1- RESOURCE ALLOCATION FAILURE.
*                        *  6- SYNCHRONIZATION FAILURE.
*                        *  7- CORRELATION FAILURE.
*  12     *  ID. OF SCARCE RESOURCE. (APPLICABLE ONLY IF WORD
*                        *  11 IS A 1 )
*  13     *  UNIQUE CLASS NUMBER.
*
*****

```

### Source Allocation Records

The source allocation records (Figure 3.5) are written out on tape or disk by the Phase 3 TRAM program each time a class of students is matriculated in a course.

A potential discrepancy can arise between the matriculation date established by subroutine SCATSA and the actual matriculation date.

SCATSA assigns classes to sources on the basis of track priorities, allocation proportions and availability of trainees. Once the assignments are made, the classes will follow the established tracks. However, if lags occur due to resource unavailability or synchronization or correlation failure, a class may reach the matriculation block at an earlier time than predicted by SCATSA.

The source allocation records are written when the class reaches the matriculation procblock and the 'GETSOURCE' task is executed.

The large time interval used for trainee inventories should serve to minimize this possible problem.

Figure 3.5

```

*****
*
*           S O U R C E   A L L O C A T I O N   R E C O R D .
*
*****
*           *
*  WORD    *  D E S C R I P T I O N
*           *
*****
*           *
*    1     *  2
*    2     *  CLASS ADRESS
*    3     *  CURRENT CLASS TIME (NOT INCLUDING DURATION OF
*           *  PROCBLOCK).
*    4     *  COURSE NUMBER
*    5     *  PROCBLOCK NUMBER.
*    6     *  TASK NUMBER.
*    7     *  SOURCE NUMBER.
*    8     *  NUMBER OF TRAINEES.
*    9     *  DURATION OF CURRENT PROCBLOCK.
*   10     *  GRADUATION DATE.
*   11     *  0
*   12     *  0
*   13     *  UNIQUE CLASS NUMBER
*           *
*****

```

#### 3.4      Subprogram Descriptions

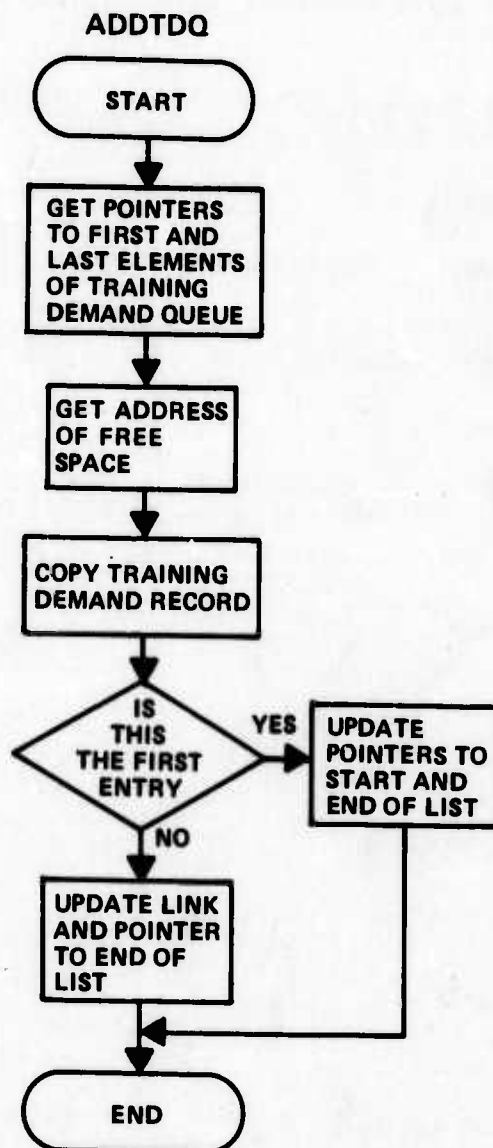
This section contains the descriptions of the individual subroutines that comprise Phase 3 of the TRAM program. The description for each subprogram consists of a statement of the purpose of the routine, the calling sequence, a description of its parameters (if any), the method used, a list of the subprograms required and the name of the programmer. A high level flowchart, which shows the logical decision points and the processing accomplished, is also included for most subprograms.

```

CC***** ADDTDQ *****
CC*
CC*  PURPOSE
CC*    UTILITY ROUTINE FOR STORING TRAINING DEMAND INFORMATION
CC*    IN A LINKED LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ADDTDQ(NCOURSE,NUMT,IDATE,IDGRAD,ITYPE,IDTYPE)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NCOURSE      NUMBER OF COURSE TO WHICH THESE TRAINEES
CC*                  SHOULD BE SENT.
CC*    NUMT         NUMBER OF TRAINEES IN THIS TRAINING
CC*                  DEMAND RECORD.
CC*                  NOTE.- VALUE IS A FLOATING POINT NUMBER.
CC*    IDATE        TRAINING DEMAND DATE.
CC*    IDGRAD       GRADUATION ID. GENERATED BY STEP 2. NOT USED.
CC*    ITYPE        TRAINEE TYPE.  1.- PILOTS
CC*                               2.- COPILOTS
CC*                               3.- OSO
CC*                               4.- DSO
CC*    IDTYPE       TRAINING DEMAND TYPE.  1.- CCTS-DELIVERY.
CC*                                           2.- CCTS-ATTRITION.
CC*                                           3.- PMT
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



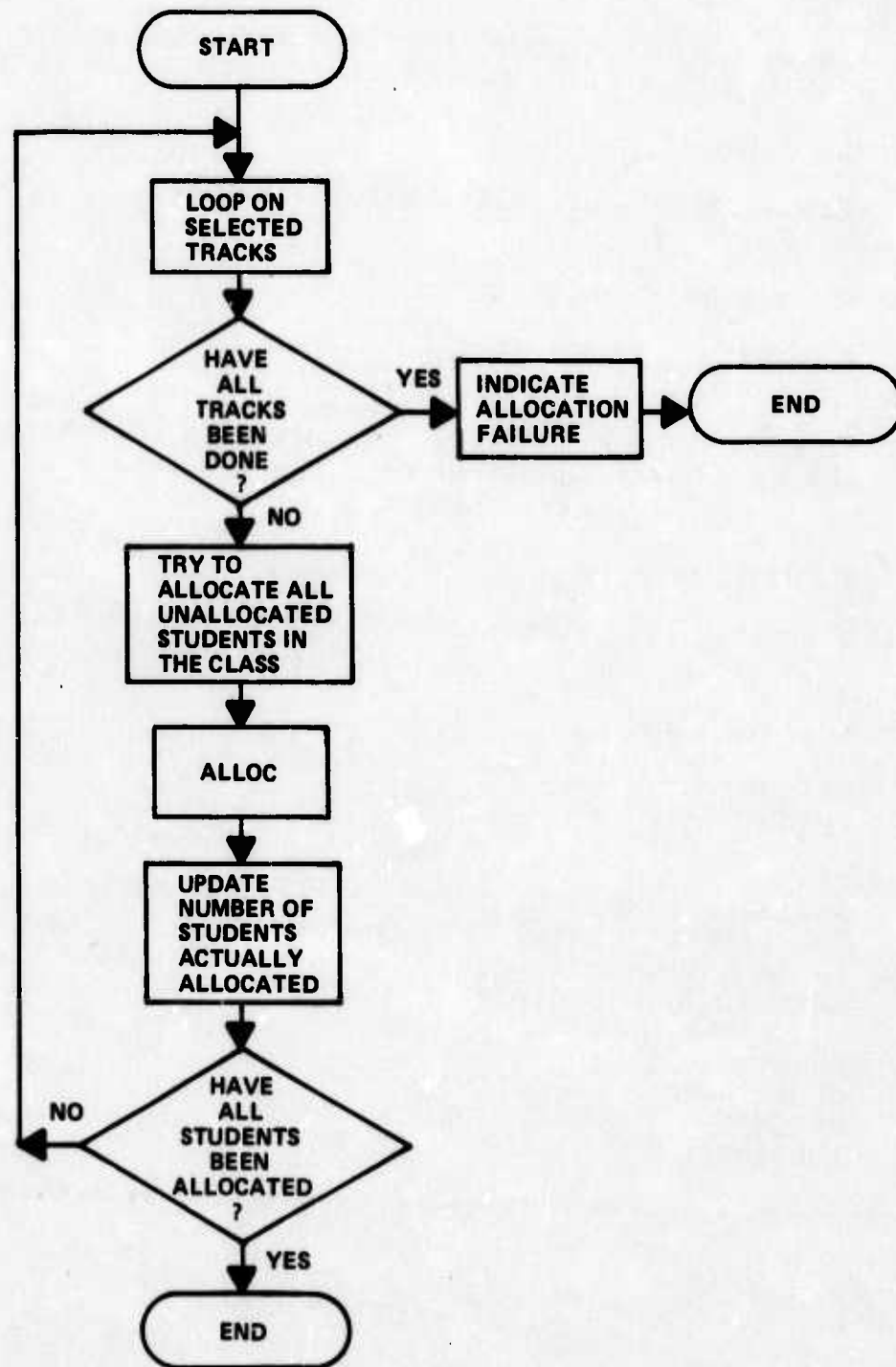


```

CC***** ALLOCA *****
CC*
CC*  PURPOSE
CC*    TO SELECT FROM A SET OF EQUAL PRIORITY TRACKS THE NUMBER
CC*    OF TRAINEES THAT HAVE NOT BEEN ALLOCATED YET.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ALLOCA(N,LIST,ICURRT,NUMSTD,  NASGND )
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    N                NUMBER OF ELEMENTS IN LIST.
CC*    LIST()          POINTERS TO TRACKS OF EQUAL PRIORITY TO
CC*                    WHICH WE WISH TO ASSIGN THIS CLASS.
CC*    ICURRT          CURRENT SIMULATION TIME.
CC*    NUMSTD          NUMBER OF STUDENTS IN THE CLASS BEING
CC*                    PROCESSED.
CC*
CC*    * I / O *
CC*
CC*    NASGND           NUMBER OF STUDENTS ALLOCATED SO FAR.
CC*
CC*  SUBROUTINES USED
CC*
CC*    ALLOC
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

# ALLOCA

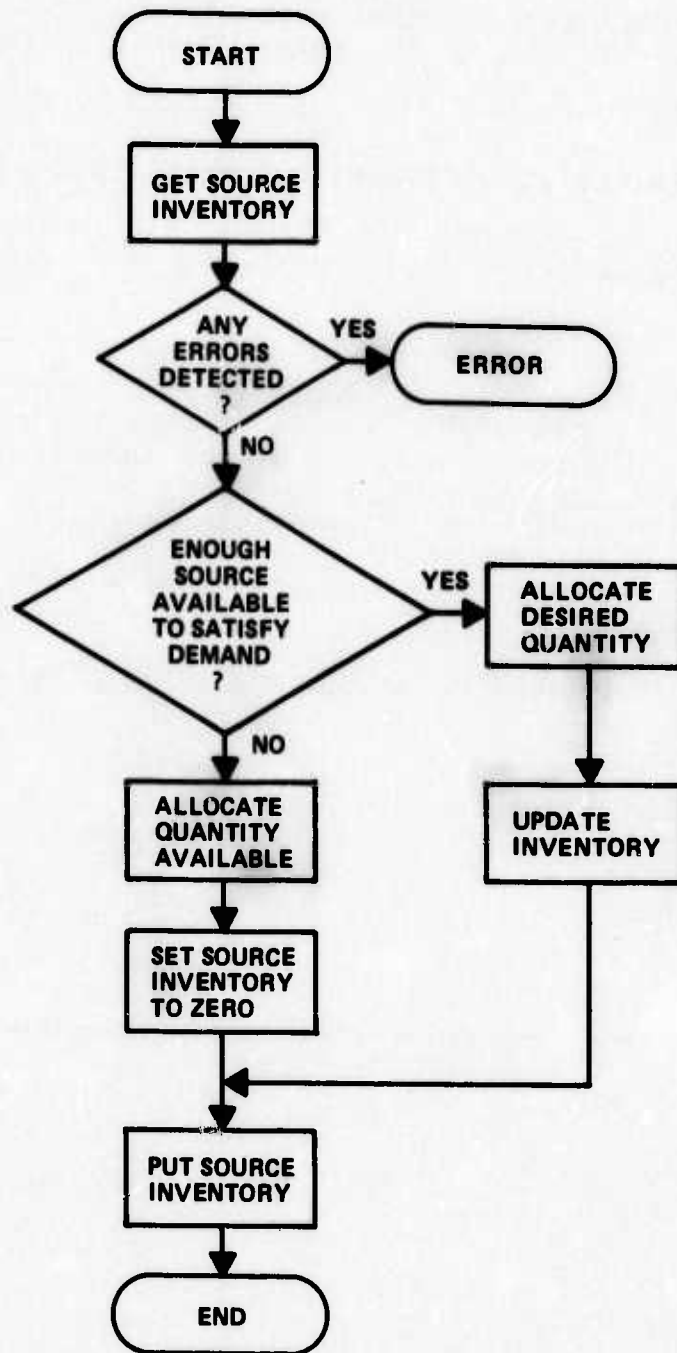


```

CC***** ALLOC *****
CC*
CC*  PURPOSE
CC*    ALLOCATE MINIMUM OF DESIRED AND AVAILABLE NUMBER OF
CC*    STUDENTS TO A GIVEN SOURCE AT A SPECIFIC PERIOD IN TIME.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ALLOC(NSORCE,IGSTME,NSTUDS,NSTUDA)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    NSORCE      NUMBER OF SOURCE
CC*    IGSTME      TIME AT WHICH SOURCE IS REQUIRED.
CC*    NSTUDS      NUMBER OF STUDENTS WE WOULD LIKE TO
CC*                MATRICULATE.
CC*
CC*    * OUTPUT *
CC*
CC*    NSTUDA      NUMBER OF STUDENTS ACTUALLY ASSIGNED TO SOURCE*
CC*
CC*  SUBROUTINES USED
CC*
CC*    GETSOR
CC*    PUTSOR
CC*
CC*  REMARKS
CC*    A SINGLE ELEMENT OF THE SOURCE INVENTORY IS LOOKED UP.
CC*    THE CELL SIZE (TIME PERIOD) FOR SOURCE INVENTORIES IS
CC*    LARGE COMPARED TO COURSE DURATION, SO THAT A PROXIMITY
CC*    SEARCH WOULD BE INAPPROPRIATE.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

# ALLOC



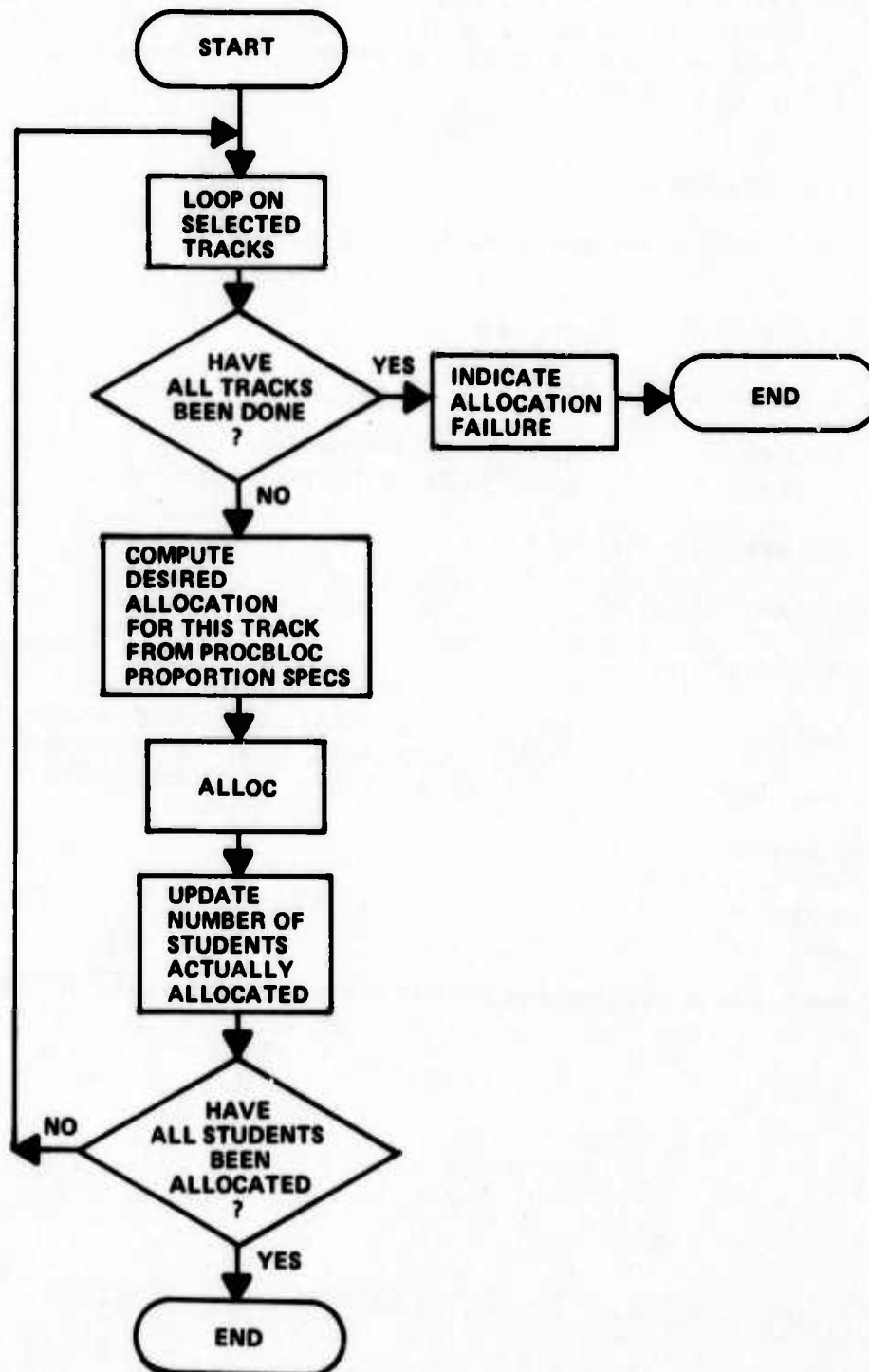


```

CC***** ALLOCD *****
CC*
CC*  PURPOSE
CC*    TO SELECT FROM A SET OF EQUAL PRIORITY TRACKS. THE NUMBER
CC*    OF TRAINEES SPECIFIED BY THE TRANSFER PROPORTIONS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ALLOCD(N,LIST,ICURRT,NUMSTD,SUMPCT, NASGND )
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    N          NUMBER OF ELEMENTS IN LIST.
CC*    LIST()     POINTERS TO TRACKS OF EQUAL PRIORITY TO
CC*               WHICH WE WISH TO ASSIGN THIS CLASS.
CC*    ICURRT     CURRENT SIMULATION TIME.
CC*    NUMSTD     NUMBER OF STUDENTS IN THE CLASS BEING
CC*               PROCESSED.
CC*    SUMPCT     SUMMATION OF PER CENTAGES - USED TO NORMALIZE
CC*
CC*    * I / O *
CC*    NASGND     NUMBER OF STUDENTS ALLOCATED TO SOURCES
CC*
CC*  SUBROUTINES USED
CC*
CC*    ALLOC
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*****

```

# ALLOCD

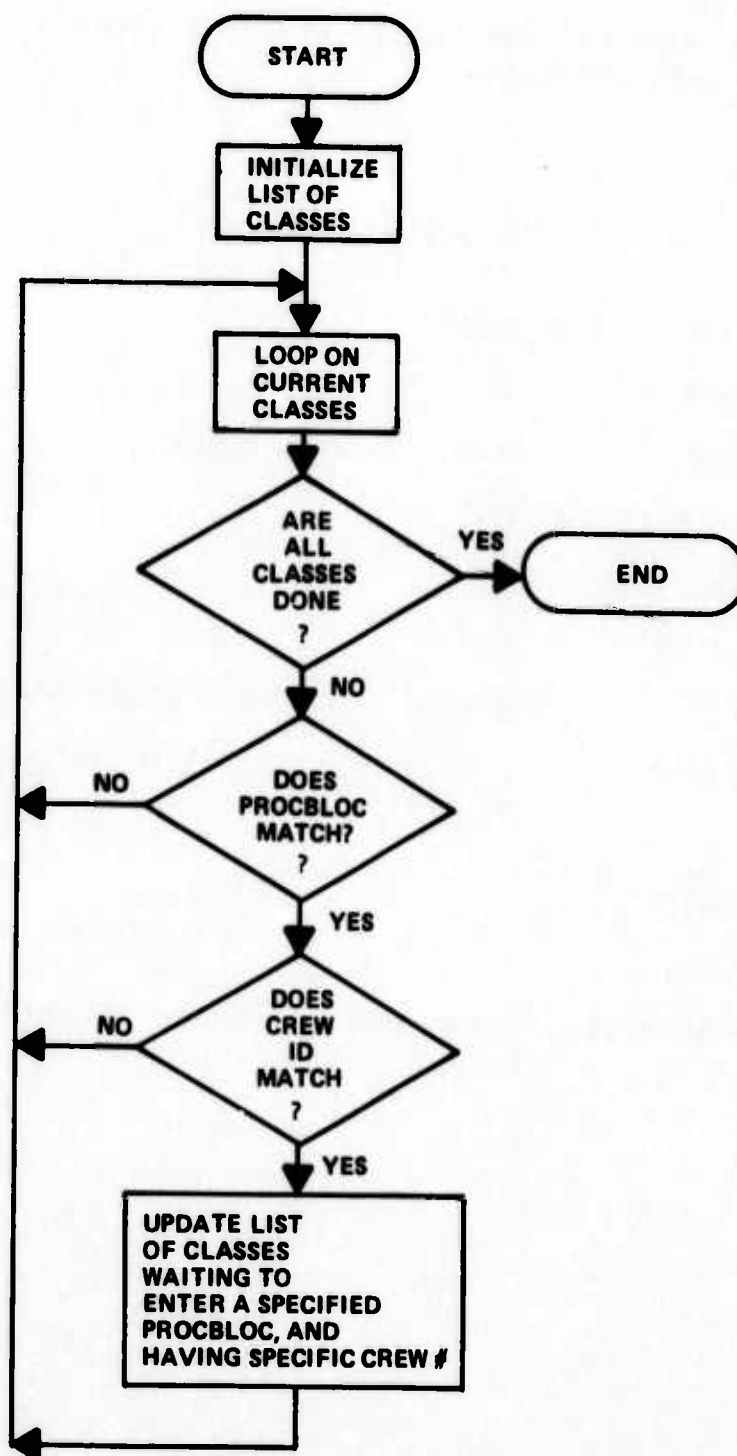


```

CC***** ASCLS *****
CC*
CC*  PURPOSE
CC*    CREATE A LIST OF ALL CLASSES WITH THE SAME GRADUATION
CC*    ID NUMBER WAITING TO ENTER A SPECIFIED PROCBLOC.
CC*    LIST CONTAINS POINTERS TO ARRAYS OF CURRENT CLASSES.
CC*    NOT CLASS ADDRESSES.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ASCLS(IPROCB,IDGRAD,NCLS,IACLS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IPROCB      ADDRESS OF PROCBLOCK
CC*    IDGRAD      CREW NUMBER (GRADUATION ID).
CC*
CC*    * IMPLICIT INPUT *
CC*
CC*    COMMON COLS
CC*
CC*    * OUTPUT *
CC*
CC*    NCLS        NO. OF CLASSES FOUND THAT MATCH THE
CC*                PROCBLOC NUMBER AND THE GRADUATION ID.
CC*    IACLS()     LIST OF POINTERS TO CLASSES IN COMMON CCLS.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# ASCLS



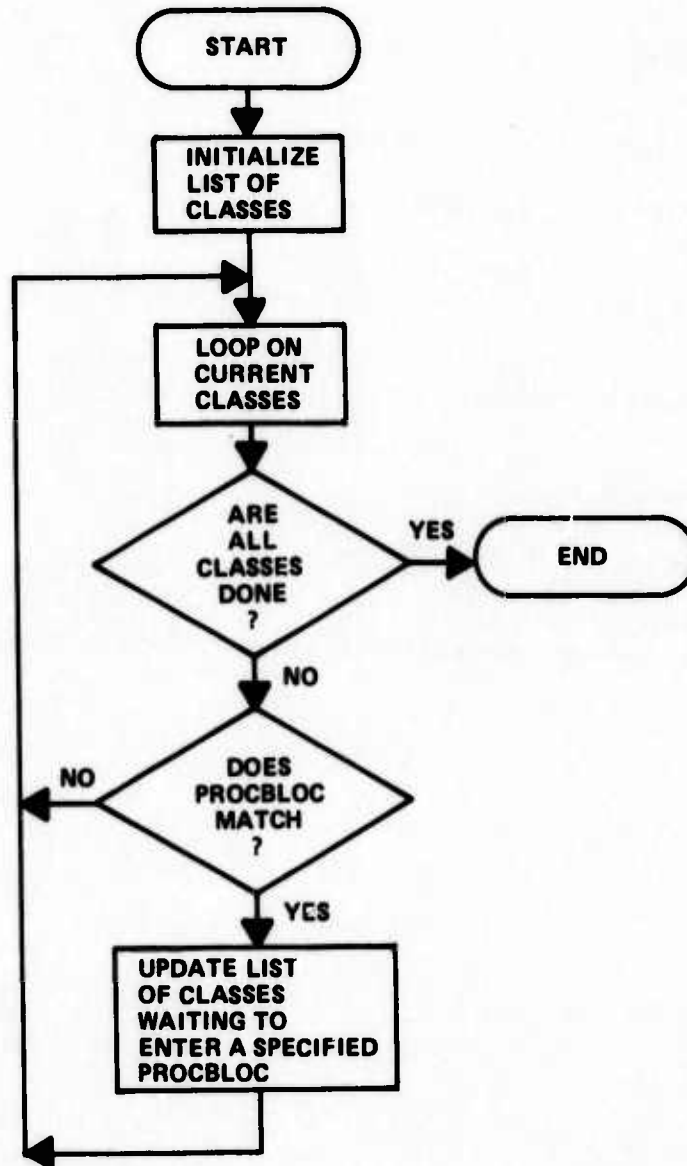
```

CC***** ASCLSS *****
CC*
CC*  PURPOSE
CC*    CREATE A LIST OF ALL CLASSES WAITING TO ENTER A SPECIFIC
CC*    PROCBLOC.
CC*    LIST CONTAINS POINTERS TO ARRAYS OF CURRENT CLASSES.
CC*    NOT CLASS ADDRESSES.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ASCLSS(IPROCB,NCLS,IACLS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IPROCB          ADDRESS OF PROCBLOCK
CC*
CC*    * IMPLICIT INPUT *
CC*
CC*    COMMON CCLS
CC*
CC*    * OUTPUT *
CC*
CC*    NCLS            NUMBER OF CLASSES ASSOCIATED WITH PROCBLOC
CC*                   AT IPROCB
CC*    IACLS()         LIST OF CLASSES ASSOCIATED WITH PROCBLOC
CC*                   AT IPROCB
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*****

```



# ASCLSS



```

C***** BLOCK *****
C*
C*          SUBROUTINE BLOCK
C*
C*  PURPOSE
C*    RETURNS THE CONTENTS OF A BLOCK
C*
C*  CALLING SEQUENCE
C*    CALL BLOCK(IADDR,IARRAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    IADDR - POINTS AT BLOCK WHOS CONTENTS IS DESIRED.
C*
C*          * EXPLICIT OUTPUT *
C*    IARRAY - CONTENTS OF BLOCK ARE PLACED IN THIS ARRAY.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    24 APRIL 1975
C*
C*****

```

```

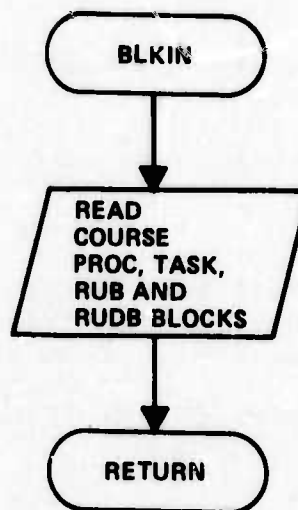
C***** BLKNAM *****
C*
C*          BLOCK DATA
C*
C*  PURPOSE
C*    INITIALIZES VARIABLE NEEDED WHEN INPUTTING NAMES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
C*****

```

```

C***** BLKIN *****
C*
C*          SUBROUTINE BLKIN
C*
C*  PURPOSE
C*    READS THE BLOCKS NEEDED TO DEFINE COURSES.
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    24 APRIL 1975
C*
C*****

```

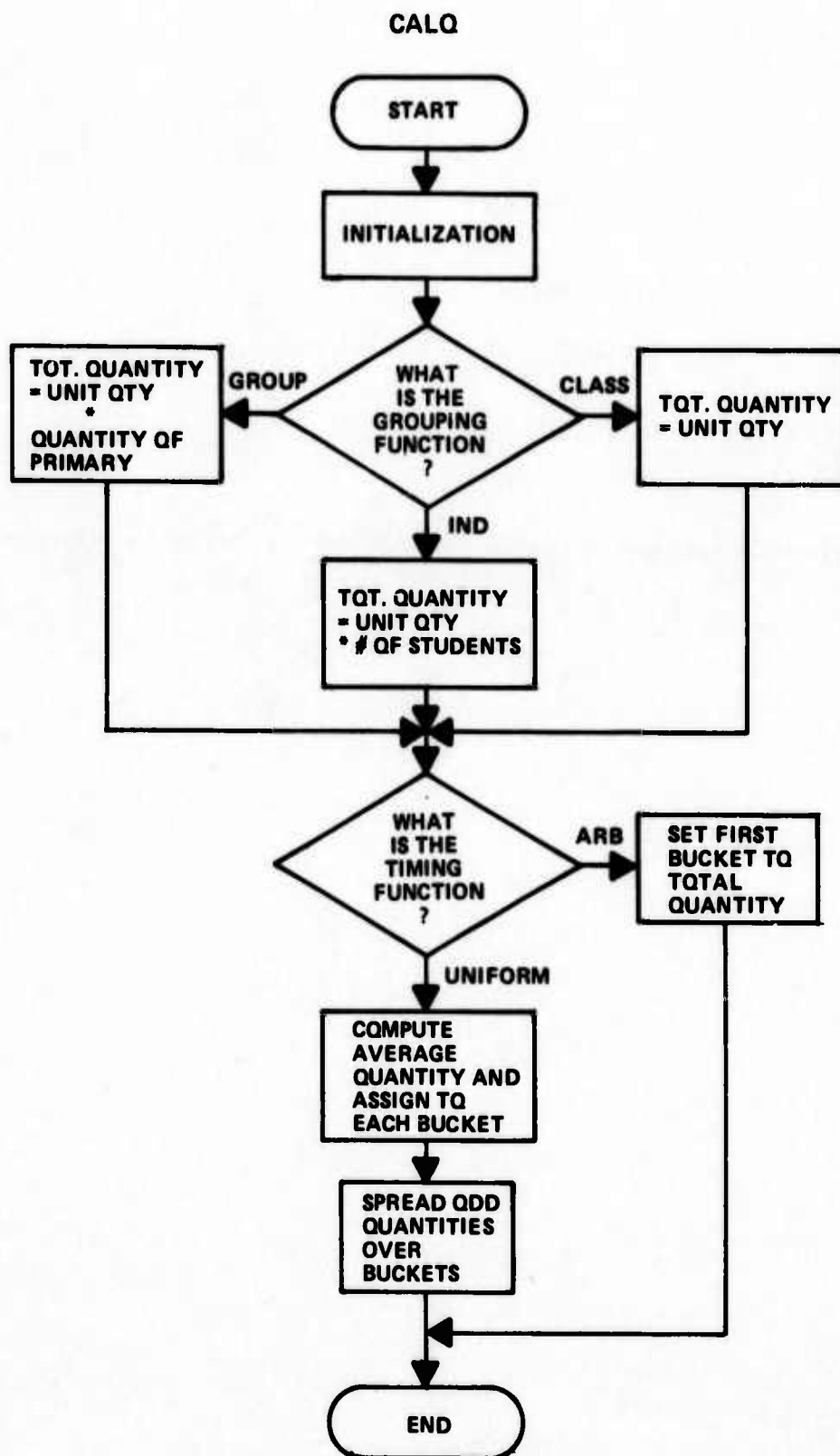




```

CC***** CALQ *****
CC*
CC*  PURPOSE
CC*    TO CALCULATE THE QUANTITY OF RESOURCE REQUIRED BY A CLASS
CC*    DOING A TASK AS A FUNCTION OF TIME.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL CALQ(INSTUDS,NRUTF,NRUGF,IQTYU,IQTYP,NBI,
CC*    1          NBO,IQTY,ITOTQ)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    INSTUDS      NUMBER OF STUDENTS IN THE CLASS
CC*    NRUTF        NUMBER OF THE RESOURCE UTILIZATION TIMING
CC*                  FUNCTION.  2- ARBITRARY
CC*                  3- UNIFORM
CC*    NRUGF        NUMBER OF THE RESOURCE UTILIZATION GROUPING
CC*                  FUNCTION.  1- CLASS
CC*                  2- QTY OF PRIMARY CONSUMED.
CC*                  3- INDIVIDUAL.
CC*    IQTYU        UNITS OF CONSUMPTION PER UNIT USER.
CC*    IQTYP        QUANTITY OF PRIMARY CONSUMED
CC*    NBI          NUMBER OF BUCKETS FROM WHICH RESOURCE
CC*                  CAN BE EXTRACTED.
CC*
CC*    * OUTPUT *
CC*
CC*    NBO          NUMBER OF BUCKETS OVER WHICH RESOURCE
CC*                  CONSUMPTION WILL BE SPREAD.
CC*    IQTY(I)      ARRAY CONTAINING THE AMOUNTS OF THE
CC*                  CONTEMPLATED RESOURCE CONSUMPTION.
CC*                  (IN REVERSED TIME SEQUENCE).
CC*    ITOTQ        TOTAL QUANTITY OF PLANNED RESOURCE
CC*                  UTILIZATION.
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

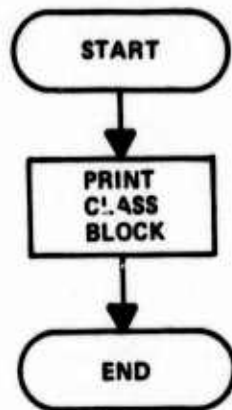


```

CC***** CBLOCK *****
CC*
CC*  PURPOSE
CC*    TO PRINT A CLASS BLOCK
CC*
CC*  PROGRAMMER
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL CBLOCK(IADRS,IBLOCK)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS      ADRESS OF CLASS BLOCK.
CC*    IBLOCK     FIRST WORD OF CLASS BLOCK
CC*
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# CBLOCK



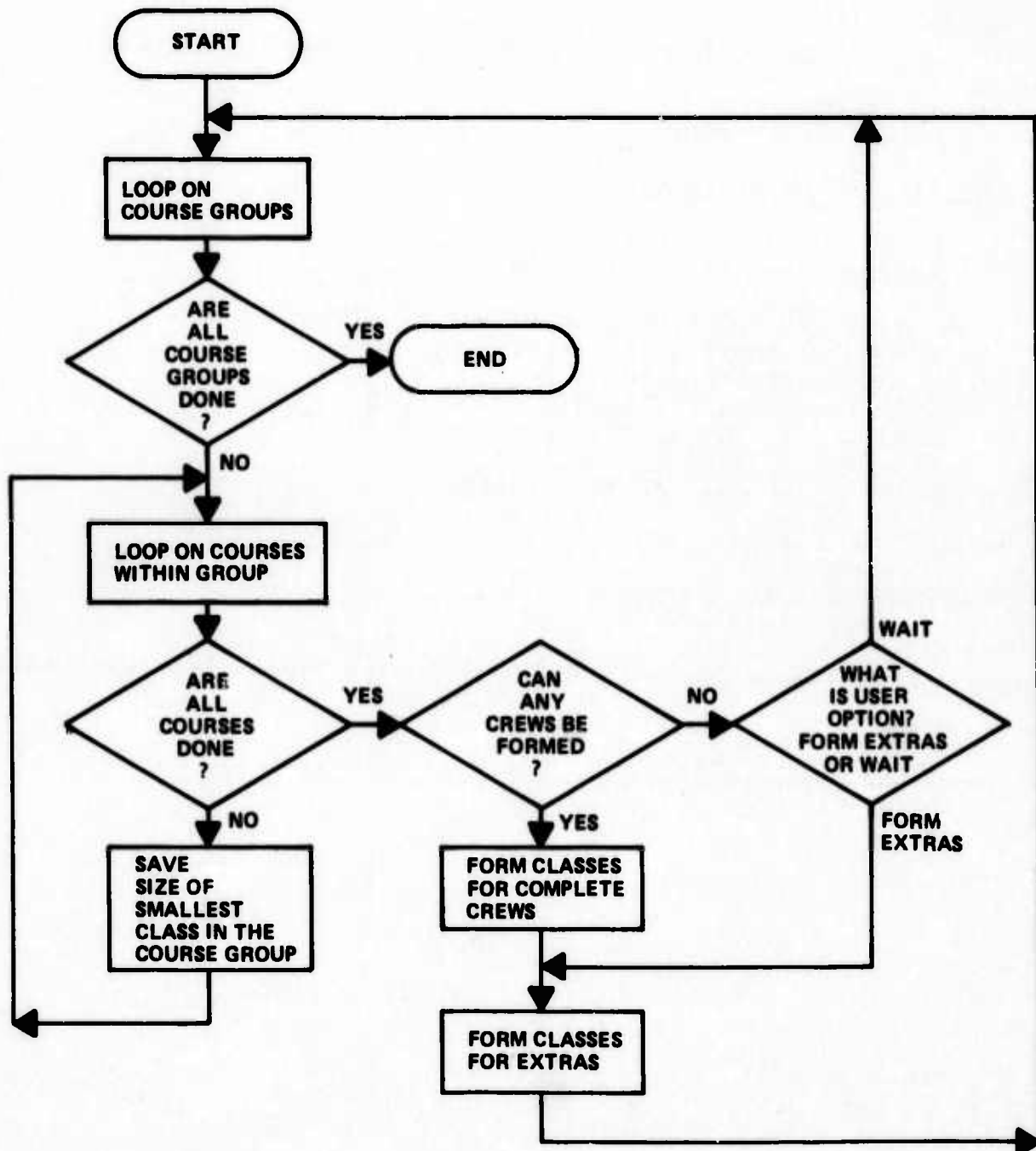
```

CC***** CLASCG *****
CC*
CC*  PURPOSE
CC*    TO FORM CLASSES FOR CREWS AND EXTRAS.
CC*    (THIS ROUTINE IS A COMPANION TO GRADF AND ASSUMES THAT
CC*    CREWS CAN BE FORMED WITHOUT REGARD TO THE DESTINATION
CC*    AIR BASE OF THE INDIVIDUAL CREW MEMBERS).
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL CLASCG
CC*
CC*  REMARKS
CC*    THIS ROUTINE LOOPS THRU ALL THE COURSES IN EACH COURSE GROUP
CC*    AND FINDS THE SMALLEST DEMAND.
CC*    IF THE DEMAND IS SUFFICIENT THEN CREW GROUPS ARE FORMED.
CC*    THE SIZE OF THE CREW IS EQUAL TO THE INTEGERIZED VALUE OF
CC*    THE SMALLEST DEMAND. EXTRAS CLASSES ARE FORMED FROM THE
CC*    REMAINING DEMANDS.
CC*    IF THE SMALLEST DEMAND IS LESS THAN ONE THEN THE PROGRAM
CC*    EITHER CREATES EXTRAS CLASSES (IOPTCG EQ 0) OR SAVES THE
CC*    DEMANDS FOR THE NEXT GRADUATION.
CC*
CC*  SUBROUTINES USED
CC*    MLTCLS
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



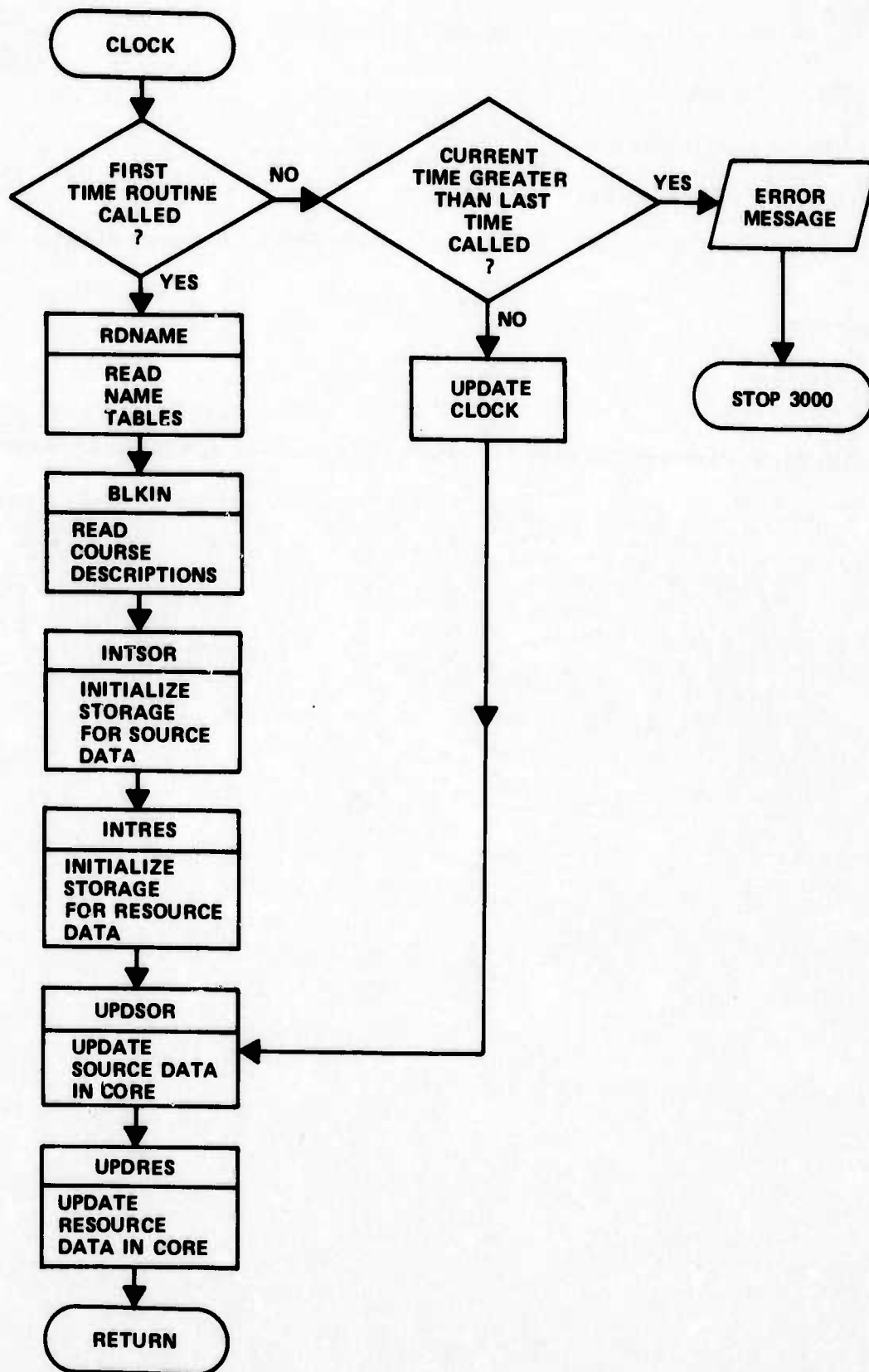
# CLASCG



```

C***** CLOCK *****
C*
C*          SUBROUTINE CLOCK
C*
C*  PURPOSE
C*    UPDATES CLOCK TIME AND UPDATES SOURCE AND RESOURCE TABLES
C*
C*  CALLING SEQUENCE
C*    CALL CLOCK(ITIME)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    ITIME - TIME TO BE ASSIGNED TO CLOCK
C*
C*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C*    RDNAME - INPUTS NAME TABLES
C*    BLKIN  - INPUTS SOURCES
C*    INTRES - INITIALIZES RESOURCE TABLES
C*    INTSOR - INITIALIZES SOURCE TABLES
C*    UPDRES - UPDATES RESOURCE TABLES
C*    UPDSOR - UPDATES SOURCE TABLES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*****

```

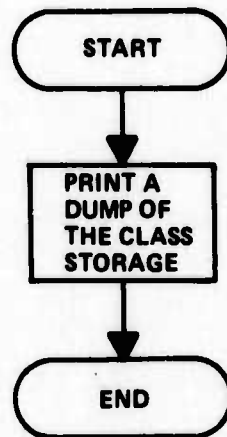


```

CC***** CLSDMP *****
CC*
CC*  PURPOSE
CC*    TO PRINT A DUMP OF THE CLASS STORAGE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL CLSDMP(IADRS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS          ADRESS OF CLASS AT WHICH TROUBLE OCCURRED.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# CLSDMP



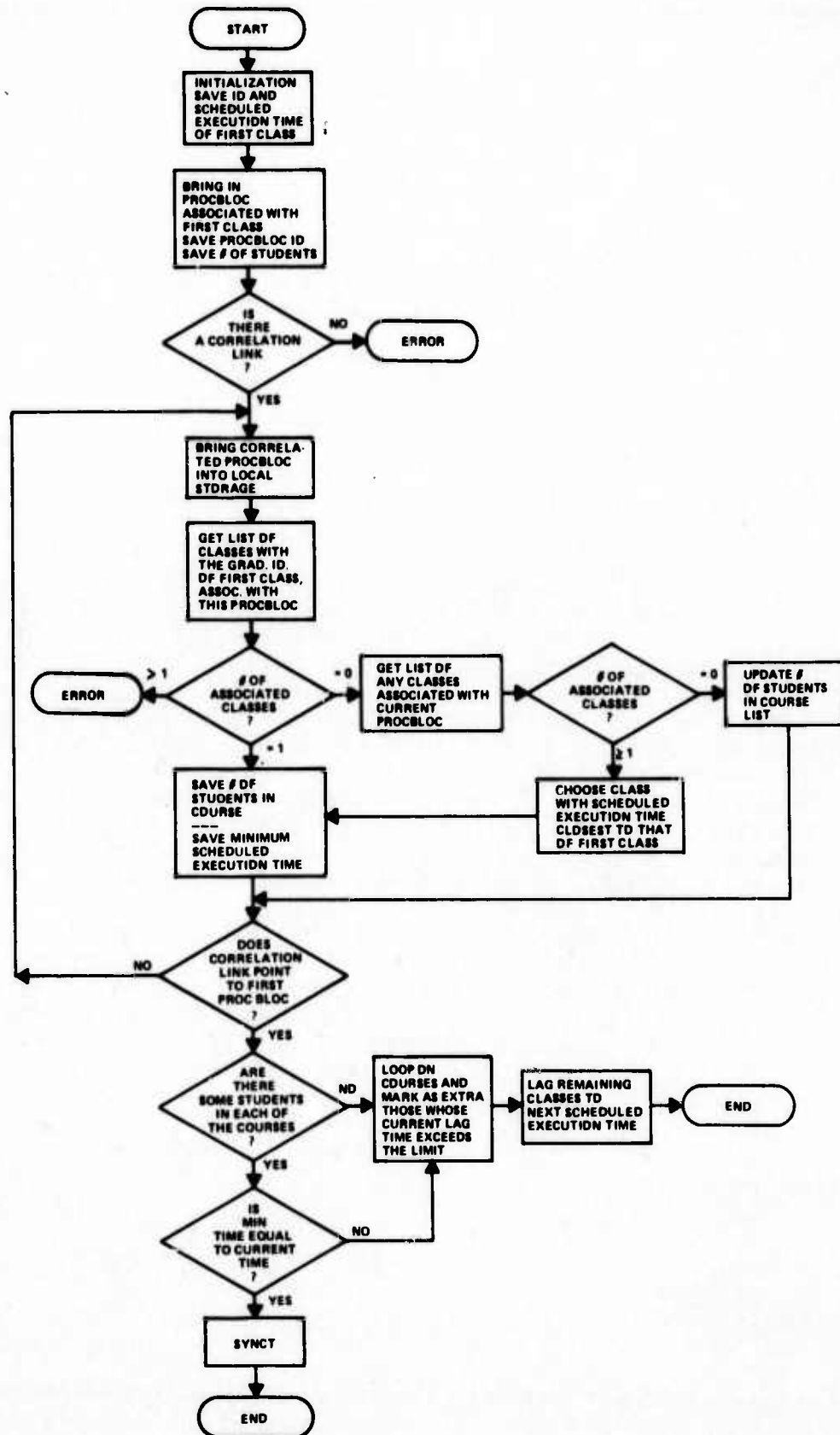


```

CC***** CORR *****
CC*
CC*  PURPOSE
CC*    TO CORRELATE THE EXECUTION OF A NUMBER OF PROCBLOCKS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL CORR(MINTME)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * OUTPUT *
CC*
CC*    MINTME          TIME TO WHICH CLASSES WILL BE LAGGED IF
CC*                   CORRELATION IS IMPOSSIBLE AT THIS TIME.
CC*
CC*    * IMPLICIT OUTPUT *
CC*
CC*    IEXTRA          IS SET TO 1, IF EXTRA TASKS ARE TO BE
CC*                   EXECUTED.
CC*    IFAIL           IS SET TO 1 IF CORRELATION CANNOT BE
CC*                   ACCOMPLISHED
CC*    NOCLS           NUMBER OF CLASSES TO BE SYNCHRONIZED
CC*                   IN EXECUTION OR LAGGED.
CC*    INDXC()         LIST OF CLASSES TO BE SYNCHRONIZED IN
CC*                   EXECUTION OR LAGGED.
CC*
CC*  REMARKS
CC*
CC*    CORRELATION MEANS THAT SOME STUDENTS ARE PRESENT IN EACH ONE
CC*    OF THE COURSES LINKED BY THE CORRELATION SPECIFICATION.
CC*    IF CORRELATION CANNOT BE ACHIEVED THE CLASSES ARE LAGGED
CC*    UP TO A MAXIMUM TIME MAXLAG. ONCE A CLASS HAS BEEN LAGGED
CC*    FOR MAXLAG TIME, IT WILL EXECUTE THE EXTRAS TASKS IF IT
CC*    CANNOT BE CORRELATED IMMEDIATELY.
CC*
CC*  SUBROUTINES USED
CC*    CLSDMP
CC*    PBLOCK
CC*    ASCLS
CC*    SYNCT
CC*    BLOCK
CC*    ASCLSS
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*****

```

# CORR

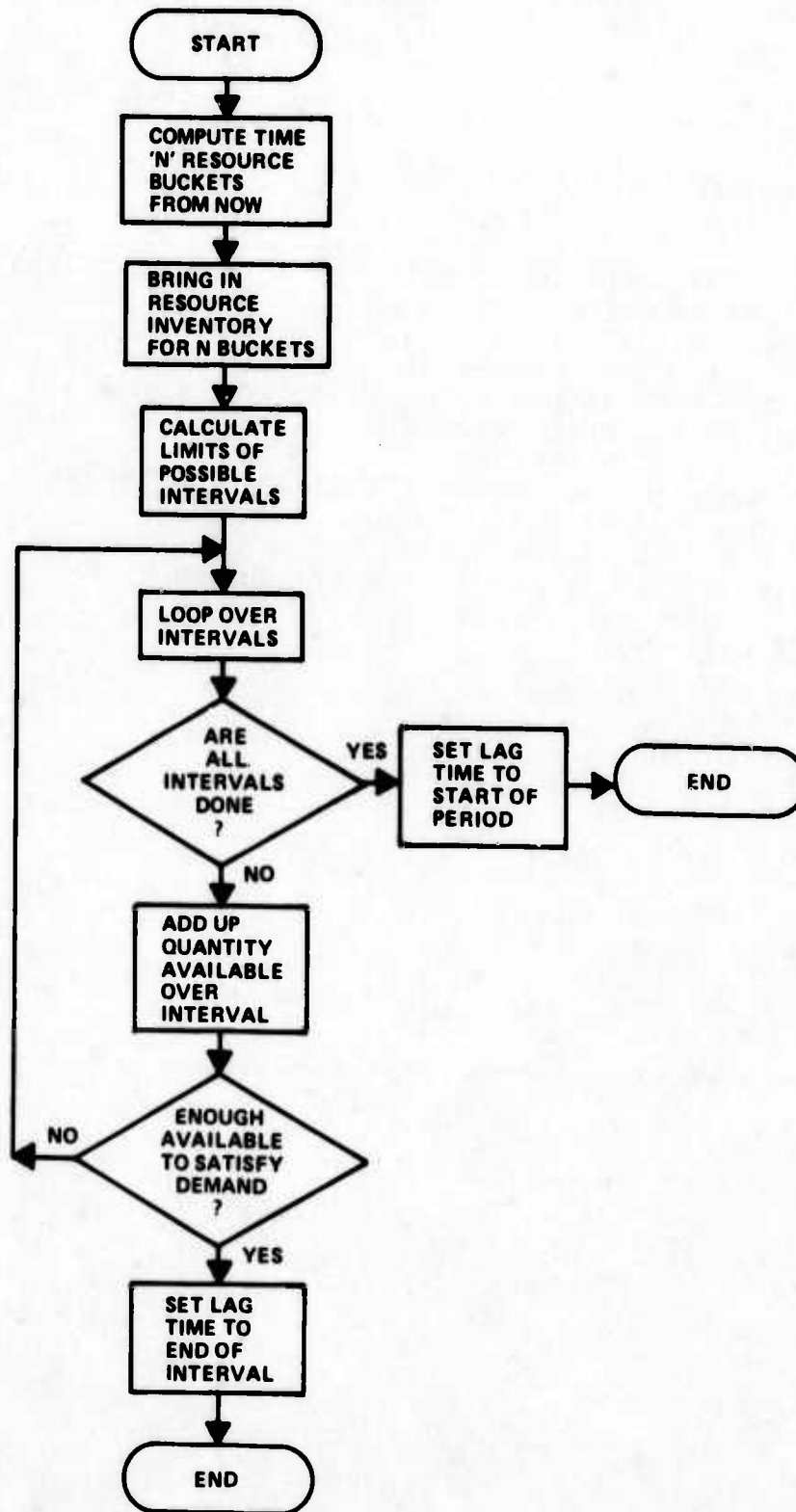


```

CC***** DETLAG *****
CC*
CC*  PURPOSE
CC*    DETERMINE HOW LONG A TIME LAG IS NECESSARY TO REACH A
CC*    PERIOD IN TIME WHEN A SPECIFIED RESOURCE IS AVAILABLE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL DETLAG(IRES,NEEDQ,LAGTME
CC*  DESCRIPTION OF PARAMETERS
CC*    IRES          NUMBER OF THE RESOURCE FOR WHICH THE
CC*                  DEMAND COULD NOT BE SATISFIED AT THIS
CC*                  TIME
CC*    NEEDQ         QUANTITY OF RESOURCE NEEDED.
CC*    LAGTME        TIME TO WHICH CLASS(ES) SHOULD BE LAGGED.
CC*
CC*  REMARKS
CC*    WHEN A DEMAND FOR RESOURCES CAN NOT BE SATISFIED DURING
CC*    THE ACTIVE TIME INTERVAL, THREE PROCESSING OPTIONS ARE
CC*    AVAILABLE:
CC*      1. STOP THE RUN.
CC*      2. INDICATE THE SHORTAGE AND CONTINUE.
CC*      3. LAG THE CLASS TO A TIME PERIOD WHEN THE
CC*         RESOURCES NEEDED ARE AVAILABLE.
CC*    THIS ROUTINE TRIES TO DETERMINE THE LATEST TIME WHEN A
CC*    RESOURCE IS AVAILABLE.
CC*    DETLAG BRINGS IN A FIXED NUMBER OF BUCKETS OF THE INVENTORY
CC*    OF THE SCARCE RESOURCE INTO LOCAL STORAGE. THIS INVENTORY
CC*    IS EXAMINED IN GROUPS OF BUCKETS CORRESPONDING TO THE
CC*    PROCBLOC INTERVAL. IF ANY PERIOD WITHIN THE TIME INTERVAL
CC*    CONSIDERED HAS ENOUGH RESOURCE AVAILABLE TO SATISFY THE
CC*    DEMAND, THE CLASS (OR CLASSES) ARE LAGGED TO THE END-TIME
CC*    OF THAT PERIOD. OTHERWISE THE CLASSES ARE LAGGED TO THE
CC*    START OF THE 'LOOK-BACK' PERIOD.
CC*    NOTES.- THE LOOK-BACK IS DONE FOR ONE CLASS-RESOURCE
CC*             COMBINATION, WITHOUT MEMORY OF OTHER RESOURCE USERS.
CC*             THEREFORE, WHEN THE CLASS TRIES TO USE THE RESOURCE
CC*             AT THE LAGGED TIME, THERE MAY NOT BE ENOUGH RESOURCE
CC*             LEFT BECAUSE ANOTHER CLASS HAS USED IT UP.
CC*             ALSO, THE CLASS MAY RUN OUT OF SOME OTHER RESOURCE
CC*             DURING THE LAGGED TIME.
CC*
CC*  SUBROUTINES USED
CC*    GETRES
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*****

```

# DETLAG



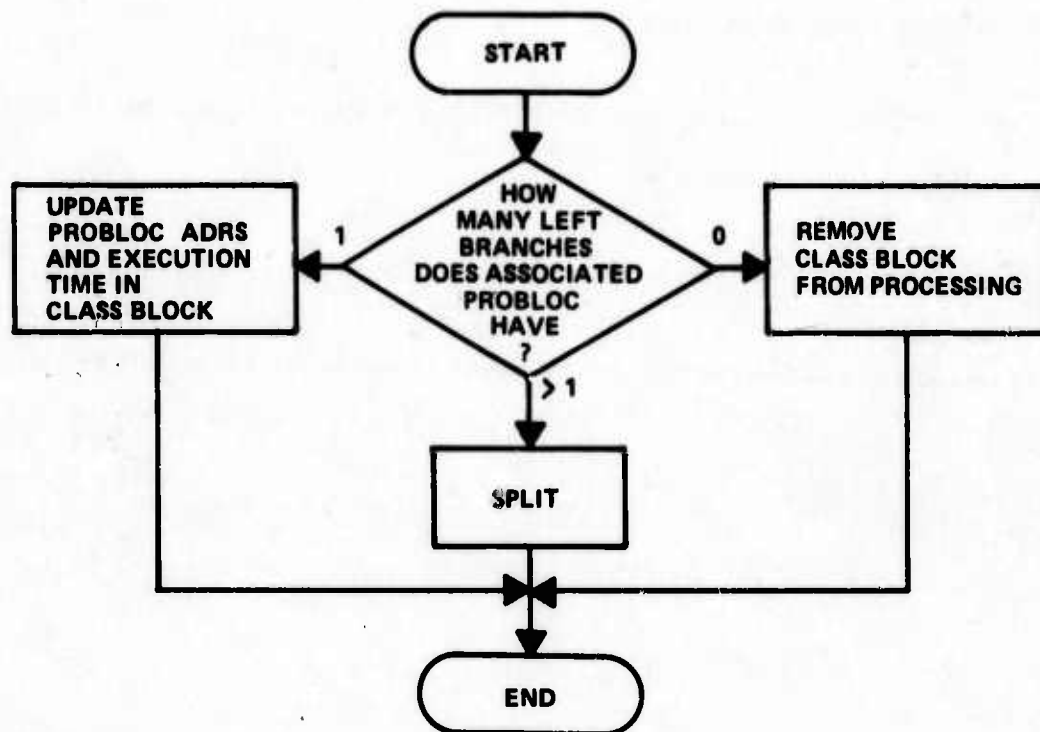
```

C***** DTRNSF *****
CC*
CC*  PURPOSE
CC*    TRANSFER A CLASS FROM ONE PROCBLOC TO THE NEXT PROCBLOC.
CC*
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL DTRNSF
CC*
CC*  REMARKS
CC*    WHEN A CLASS ENTERS DTRNSF IT MEANS THAT ALL TASKS FOR
CC*    THE ACTIVE PROCBLOC HAVE BEEN SUCCESSFULLY COMPLETED.
CC*    THE CLASS PRIORITY IS RESTORED TO EQUAL THE COURSE
CC*    PRIORITY, AND THE NEXT-BREAK TIME IS UPDATED IF APPROPRIATE.
CC*    IF THE PROCBLOC IN WHICH THE CLASS IS LOCATED HAS ONLY
CC*    ONE LEFT BRANCH THEN THE CLASS PARAMETERS ARE UPDATED IN
CC*    PLACE.
CC*    IF THE PROCBLOC HAS NO BRANCHES, THEN THE CLASSBLOCK IS
CC*    REMOVED FROM STORAGE BY ROUTINE REMCLS
CC*    IF THE PROCBLOC HAS MORE THAN ONE BRANCH THEN SUBROUTINE
CC*    SPLIT IS INVOKED TO EFFECT THE TRANSFER.
CC*
CC*  SUBROUTINES USED
CC*    BLOCK
CC*    PBLOCK
CC*    SPLIT
CC*    PUTCLS
CC*    CBLOCK
CC*    REMCLS
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



DTRNSF

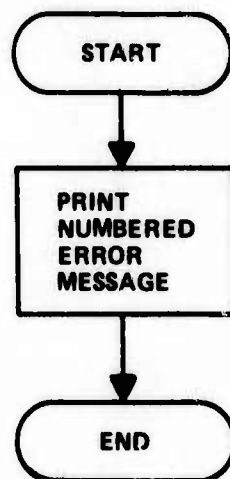


```

CC***** ERROR *****
CC*  PURPOSE
CC*    TO PRINT ERROR MESSAGES.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL ERROR(N,NAME)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*      N          ERROR NUMBER
CC*      NAME(2)    NAME OF SUBROUTINE IN WHICH ERROR OCCURRED.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

## ERROR

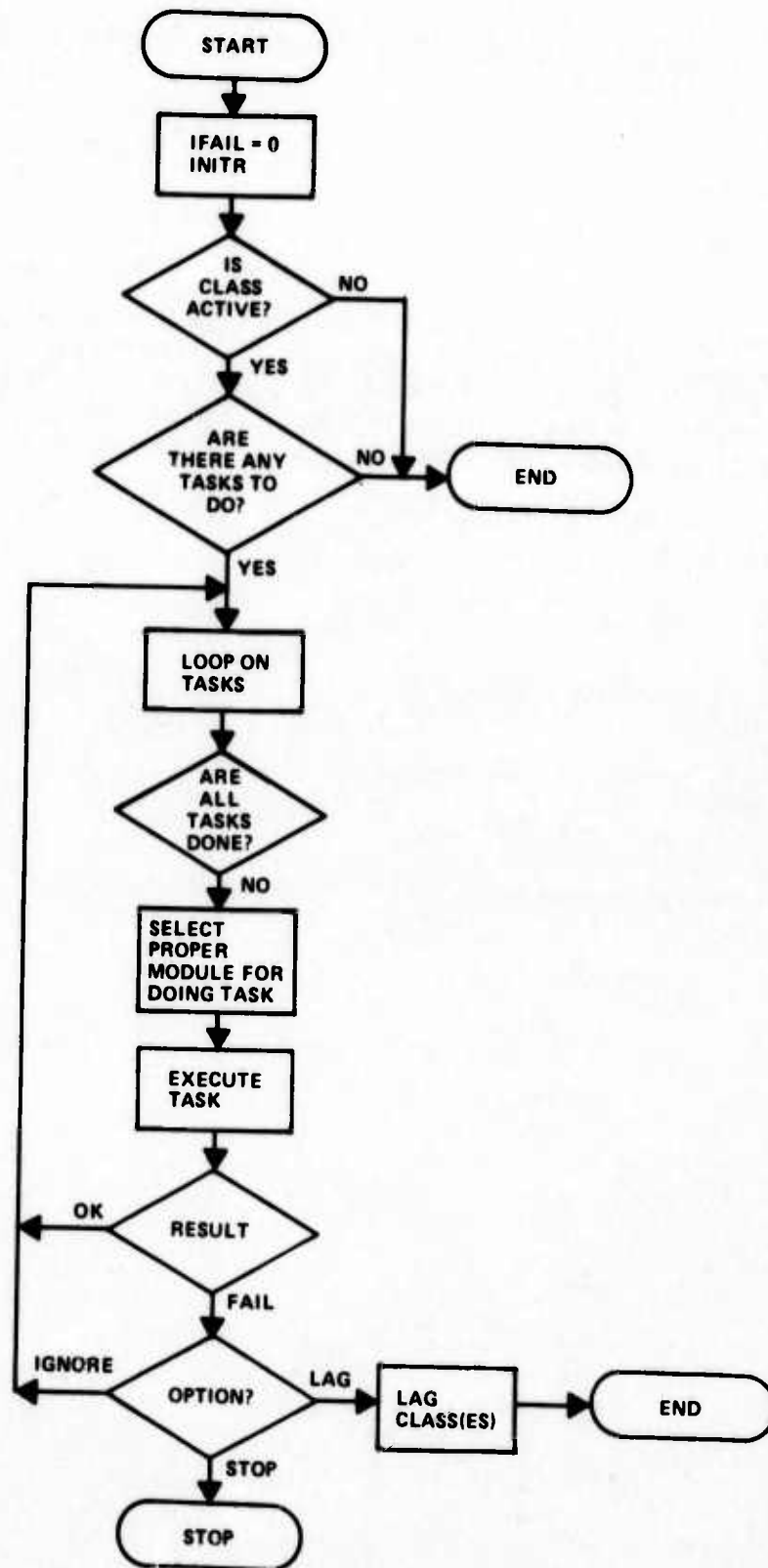


```

CC***** EXECT *****
CC*
CC*  PURPOSE
CC*    TO EXECUTE THE TASKS DESCRIBED IN THE TASK LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL EXECT
CC*
CC*  REMARKS
CC*
CC*    THIS ROUTINE CONTROLS THE EXECUTIONS OF THE TASKS WITHIN
CC*    PROCBLOCKS BY CLASSES. AFTER INITIALIZATION, EXECT INVOKES
CC*    SUBROUTINE LSTASK TO ARRANGE THE TASKS IN PROPER ORDER
CC*    AND SUPPLY THE NECESSARY PROGRAM SUPPLIED TASKS (I.E. UPDATE,
CC*    DTRNSF). THEN EXECT LOOPS THRU THE TASKS AND INVOKES THE
CC*    PROPER ROUTINES TO PERFORM THE TASK. NOTE THAT SYNC AND
CC*    CORR CHANGE THE LIST OF TASKS SO THAT THE LOOP MUST BE
CC*    RESTARTED.
CC*    IF A TASK FAILS THREE OPTIONS ARE AVAILABLE TO THE USER.
CC*    THE OPTIONS ARE STOP,IGNORE OR LAG THE CLASSES.
CC*
CC*  SUBROUTINES USED
CC*    CORR
CC*    DTRNSF
CC*    RESUSE
CC*    INTR
CC*    GETCLS
CC*    SCATSA
CC*    SYNC
CC*    PLIST
CC*    TBLOCK
CC*    LAG
CC*    BLOCK
CC*    CBLOCK
CC*    UPDATE
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# EXECT



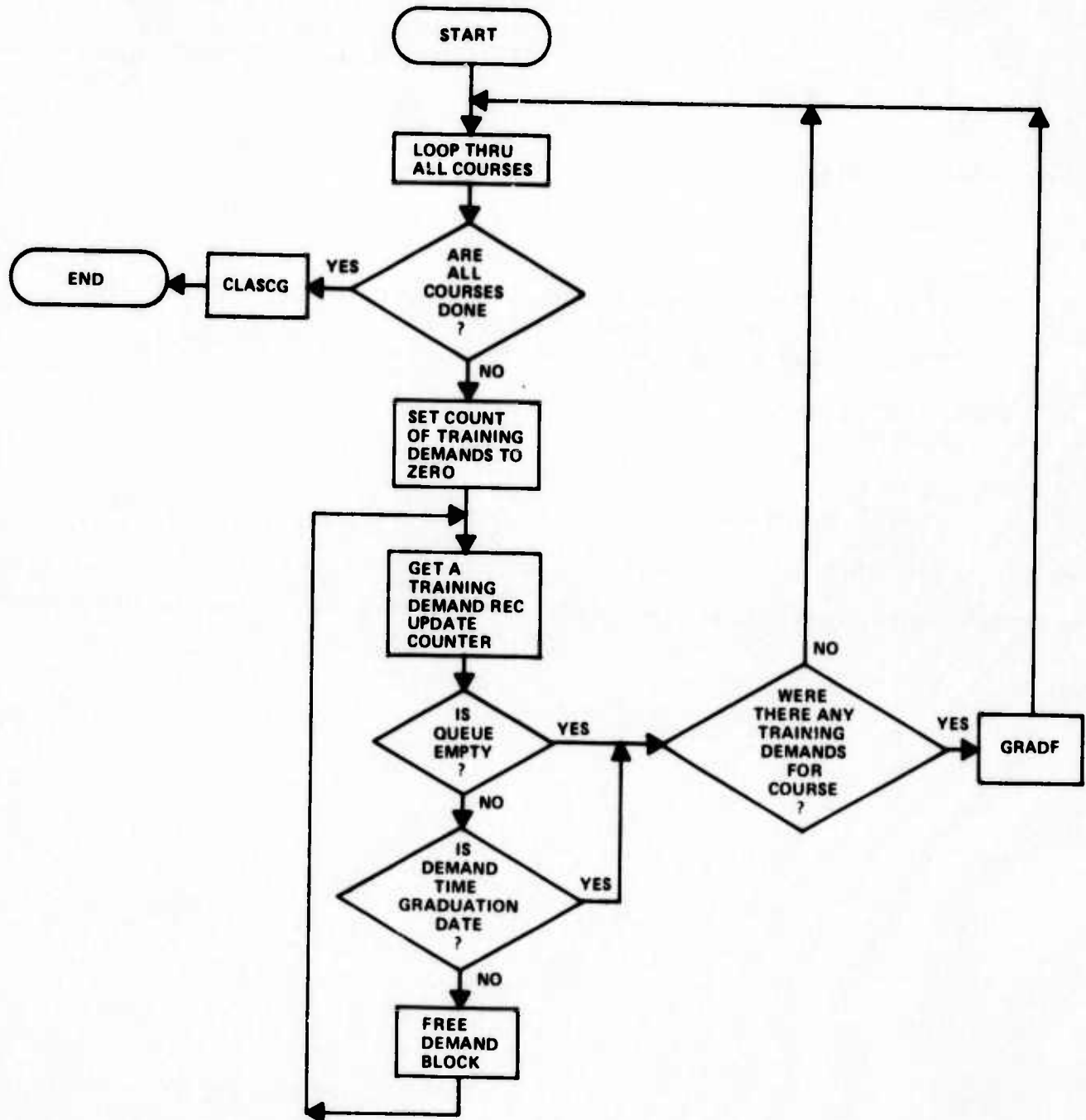


```

CC***** FORMC *****
CC*
CC*  PURPOSE
CC*    CREATE CLASS BLOCKS FROM THE TRAINING DEMAND INFORMATION
CC*    STORED BY 'FORMQ'.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL FORMC
CC*
CC*  REMARKS
CC*    THIS ROUTINE EXTRACTS FROM THE TRAINING DEMAND QUEUE
CC*    THE RECORDS NECESSARY TO COMPUTE THE GRADUATION
CC*    REQUIREMENTS FOR EACH COURSE. THE STORAGE OCCUPIED BY
CC*    THE TRAINING DEMANDS IS RETURNED TO FREE SPACE.
CC*    SUBROUTINES GRADF AND CLASCG ARE USED TO FORM THE
CC*    CLASSES
CC*
CC*  SUBROUTINES USED
CC*    CLASCG
CC*    FRETDB
CC*    GETTDB
CC*    GRADF
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# FORMC

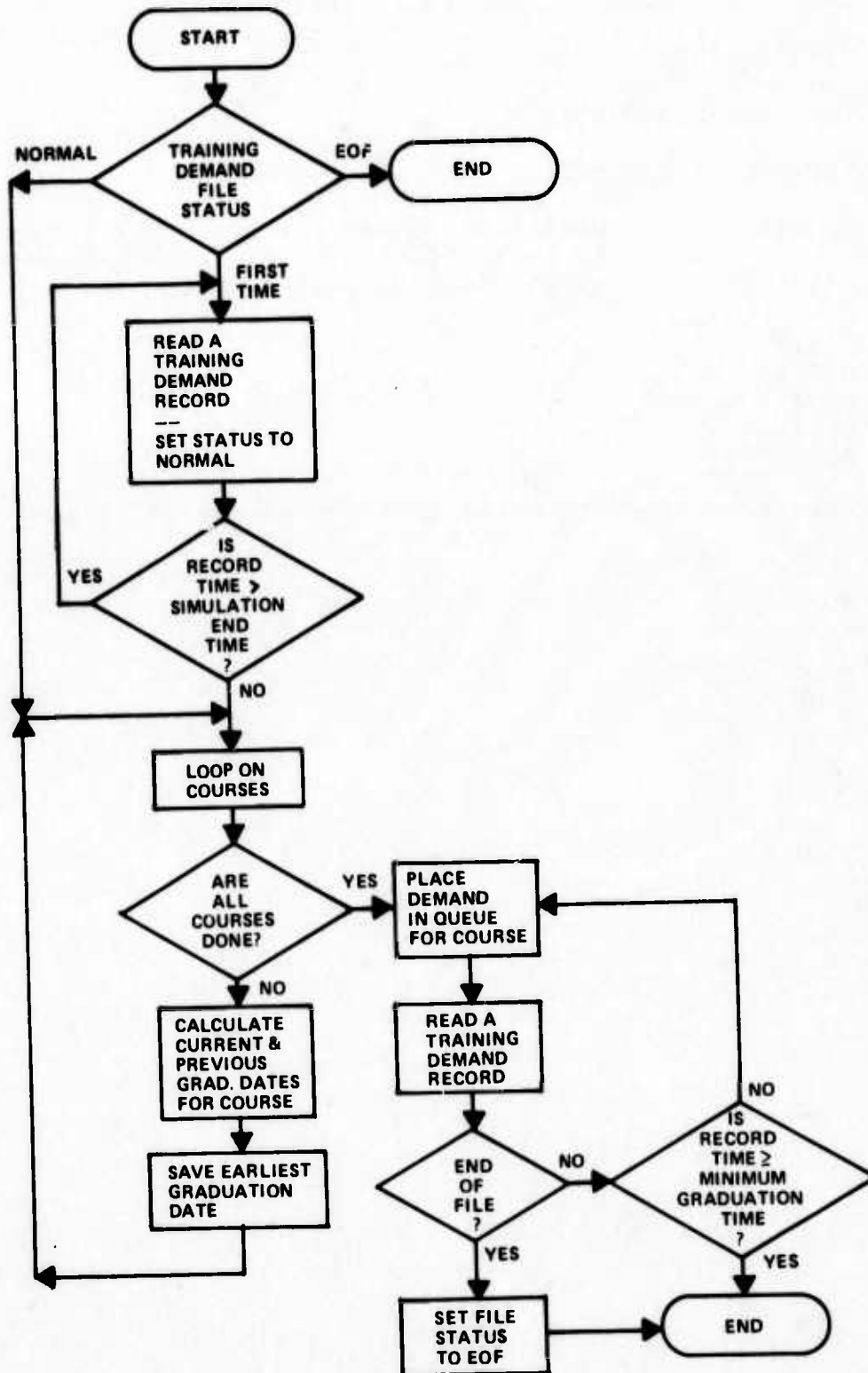


```

CC***** FORMQ *****
CC*
CC*  PURPOSE
CC*    READ IN TRAINING-DEMAND RECORDS GENERATED BY TRAM2 FOR
CC*    A PERIOD OF TIME TO MEET CURRENT GRADUATION SCHEDULES
CC*    FOR ALL COURSES AND FORM INDIVIDUAL TRAINING DEMAND QUEUES
CC*    FOR EACH COURSE
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL FORMQ
CC*
CC*  REMARKS
CC*    TRAINING DEMAND RECORDS ARE READ ONLY TO THE TIME OF THE
CC*    LATEST GRADUATION AMONG THE COURSES.
CC*    THE RECORDS ARE STORED IN A ONE-DIMENSIONAL LINKED LIST.
CC*
CC*  SUBROUTINES USED
CC*    ADDTDQ
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# FORMQ



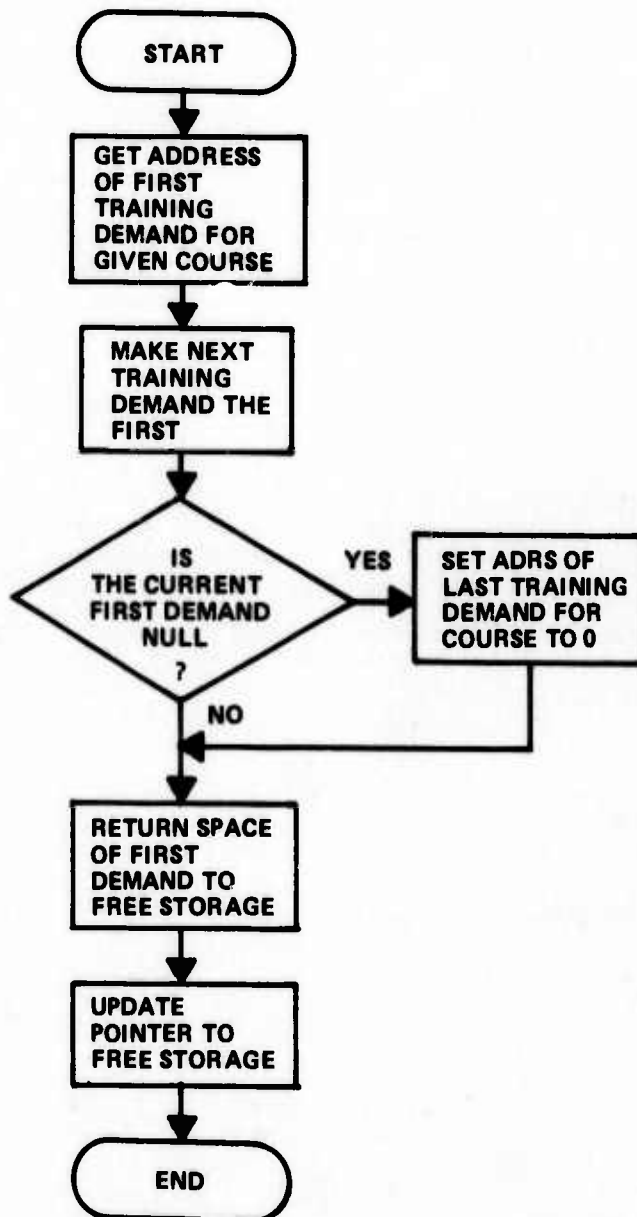
```

CC***** FRETDB *****
CC*
CC*  PURPOSE
CC*    UTILITY ROUTINE FOR FREEING STORAGE NO LONGER USED BY
CC*    TRAINING DEMAND RECORDS IN A LINKED LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL FRETDB(NCORSE)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NCORSE          NUMBER OF COURSE FOR WHICH THE CORE OCCUPIED
CC*                   BY THE FIRST TRAINING DEMAND RECORD WILL
CC*                   BE RETURNED TO FREE SPACE.
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



# FRETDB

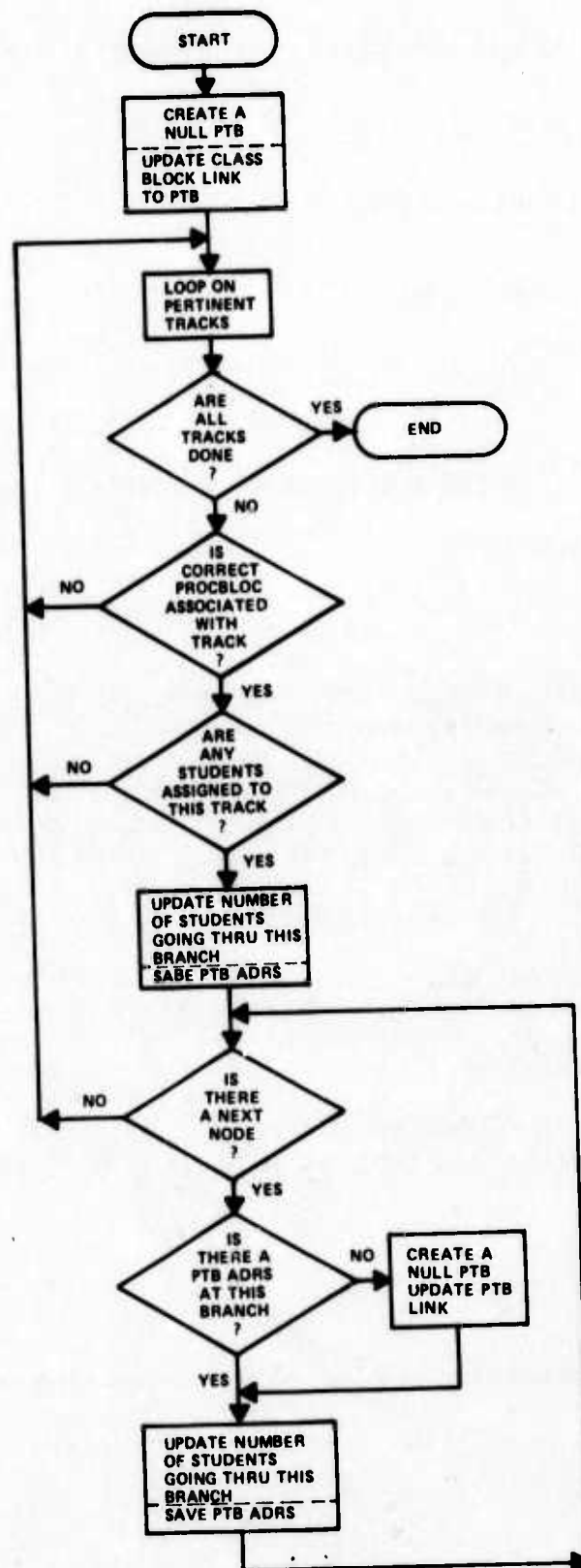


```

CC***** FRMPTB *****
CC*
CC*  PURPOSE
CC*    TO CREATE THE PREDETERMINED TRANSFER BLOCKS FOR A CLASS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL FRMPTB(L1,L2)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    L1          POINTER TO FIRST TDB FOR COURSE.
CC*    L2          POINTER TO LAST TDB FOR COURSE.
CC*
CC*  SUBROUTINES USED
CC*
CC*    PUTPTB
CC*
CC*  REMARKS
CC*    PROCESSING IN THIS ROUTINE CONSISTS OF CREATING A PTB
CC*    FOR THE PROCBLOC FROM WHICH SCATSA WAS CALLED.
CC*    THIS PTB INDICATES HOW MANY STUDENTS WILL BE GOING THRU
CC*    EACH BACKWARD BRANCH. THE NODE CHAINS FROM THE TRACK
CC*    DESCRIPTOR BLOCKS ARE FOLLOWED FOR EACH BRANCH, AND
CC*    PTBS ARE CREATE FOR EACH NODE IN THE TRACKS LEADING TO
CC*    THE SELECTED SOURCES.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

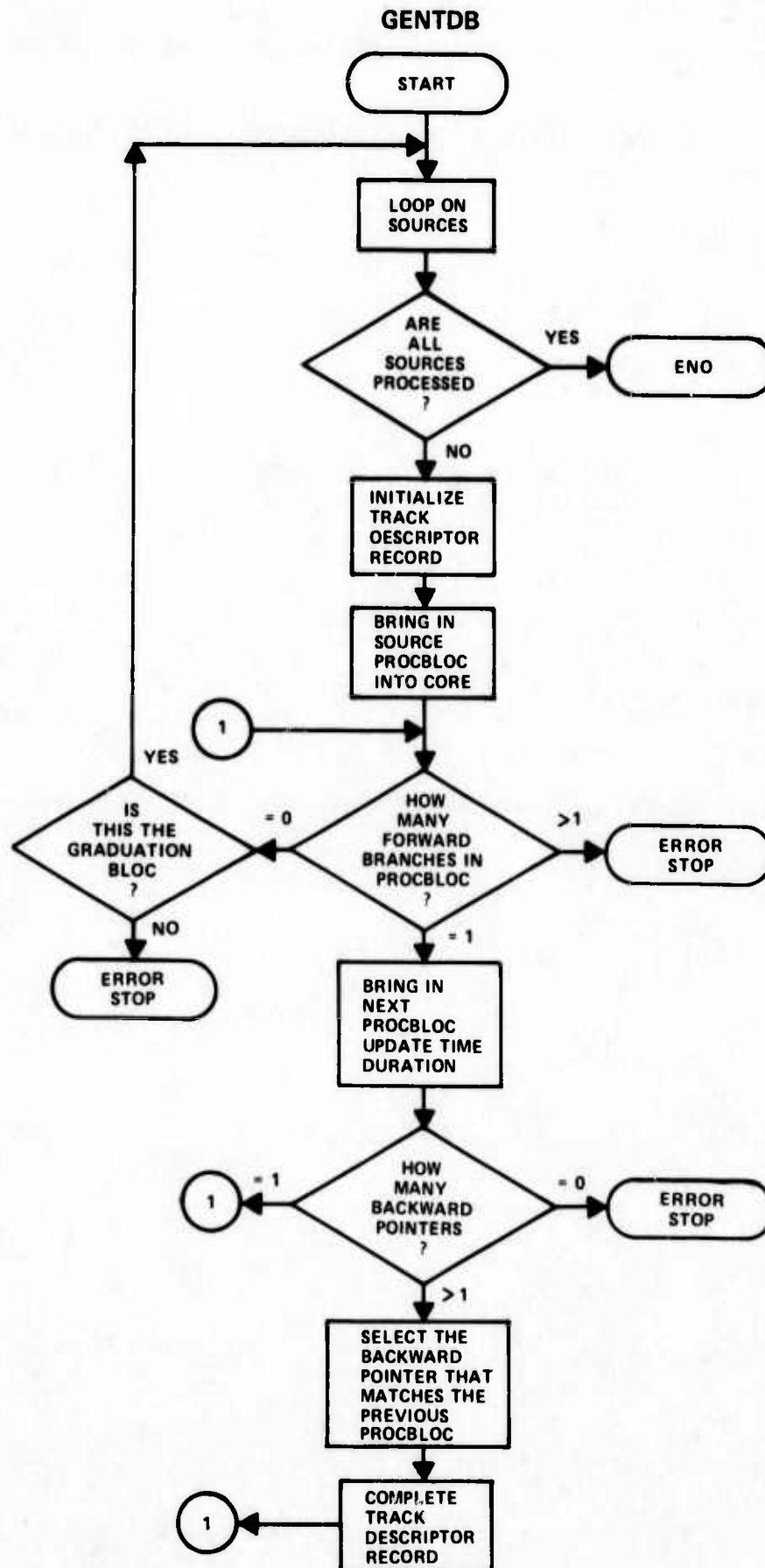
# FRMPTB



```

CC***** GENTDB *****
CC*
CC*  PURPOSE
CC*    TO GENERATE A SET OF TRACK DESCRIPTOR BLOCKS FOR EACH
CC*    COURSE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL GENTDB(ISRCE1,IDUMP )
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    ISRCE1      POINTER TO FIRST SOURCE FOR THIS COURSE.
CC*                (NOSRCS POINTS TO THE LAST SOURCE).
CC*    IDUMP       DIAGNOSTIC DUMP SWITCH. (1-PRINT DIAGS.)
CC*
CC*  SUBROUTINES CALLED
CC*    BLOCK
CC*    PBLOCK
CC*
CC*  REMARKS
CC*    GENTDB IS CALLED ONCE FOR EACH COURSE. ISRCE1 POINTS TO THE
CC*    FIRST SOURCE FOR THE COURSE, NOSRCS IN COMMON SORDSC POINTS
CC*    TO THE LAST SOURCE FOR THE COURSE. THE ROUTINE STARTS AT
CC*    EACH SOURCE AND USING THE FORWARD POINTERS STEPS THRU THE
CC*    PROCBLOCKS UNTIL A NODE IS FOUND.(PROCBLOCK WITH MORE THAN
CC*    ONE BACKWARD POINTER). CUMULATIVE TIME OF THE PROCBLOCKS,
CC*    FROM THE SOURCE TO THE NODE (INCLUSIVE) IS CALCULATED
CC*    AS ARE THE CUMULATIVE PRIORITIES AND PERCENTAGES (PRESENT
CC*    ONLY AT NODES).
CC*    A DESCRIPTION OF EACH NODE OF EVERY TRACK IS STORED IN THE
CC*    ARRAYS OF COMMON RLTDSC. EACH TRACK DESCRIPTOR BLOCK
CC*    POINTS TO THE PRECEEDING NODE OF THE SAME TRACK.
CC*    VARIABLE NSRCE POINTS TO THE SOURCE DESCRIPTION IN COMMON
CC*    BLOCK SORDSC.
CC*    NOTE.- TRACK DESCRIPTOR BLOCKS ARE CREATED ONLY FOR
CC*           COURSES THAT CONTAIN MORE THAN 1 TRACK.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*****

```



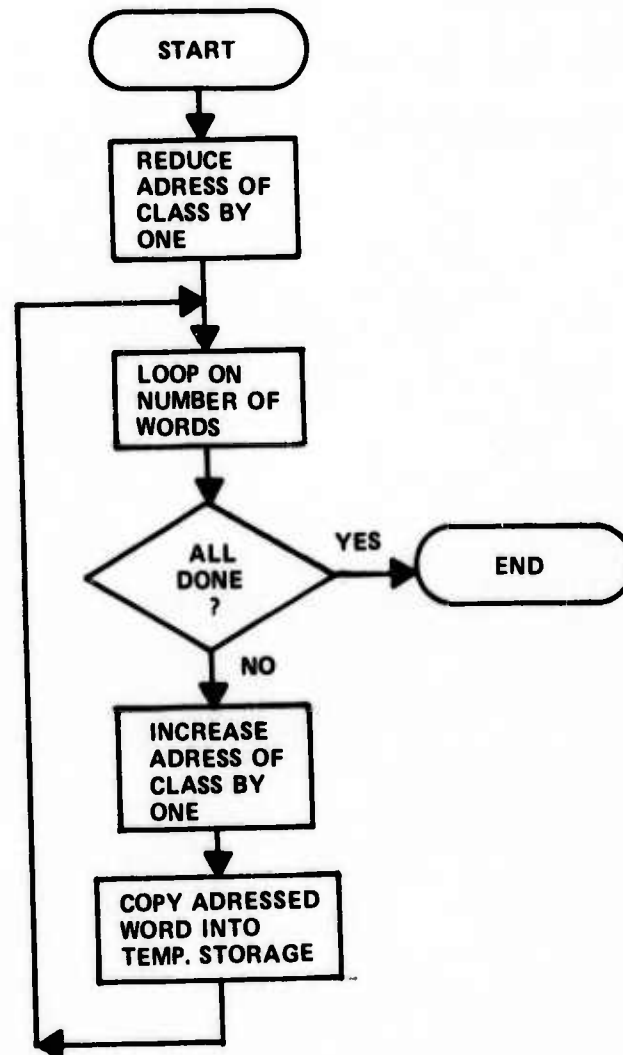


```

CC***** GETCLS *****
CC*
CC*  PURPOSE
CC*    TO MOVE A VARIABLE NUMBER OF WORDS FROM A CLASS BLOCK
CC*    INTO LOCAL STORAGE
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL GETCLS(INDEX,IA,N)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    INDEX          ADRESS OF THE CLASS BLOCK
CC*    N              NUMBER OF WORDS TO BE MOVED TO LOCAL STORAGE
CC*
CC*    * OUTPUT *
CC*
CC*    IA()           N WORDS OF CLASS BLOCK
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# GETCLS

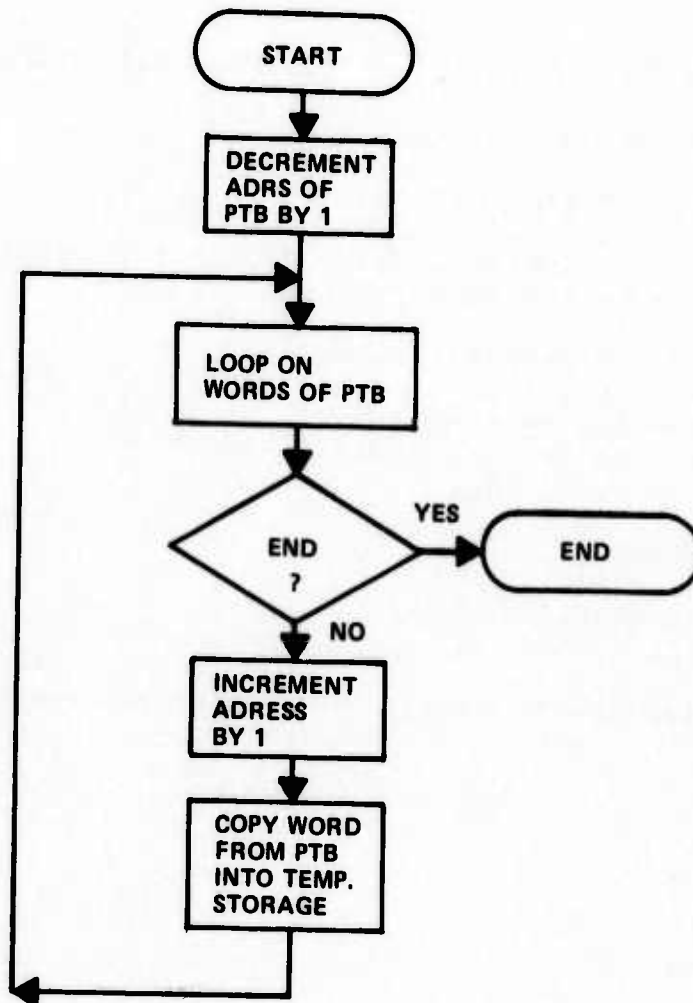


```

CC***** GETPTB *****
CC*
CC*  PURPOSE
CC*    TO BRING IN A PREDETERMINED TRANSFER BLOCK INTO LOCAL
CC*    STORAGE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL GETPTB(IADRS,IA)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IADRS          ADRESS OF PTB
CC*
CC*    * OUTPUT *
CC*
CC*    IA()          TEN WORDS OF PTB
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# GETPTB

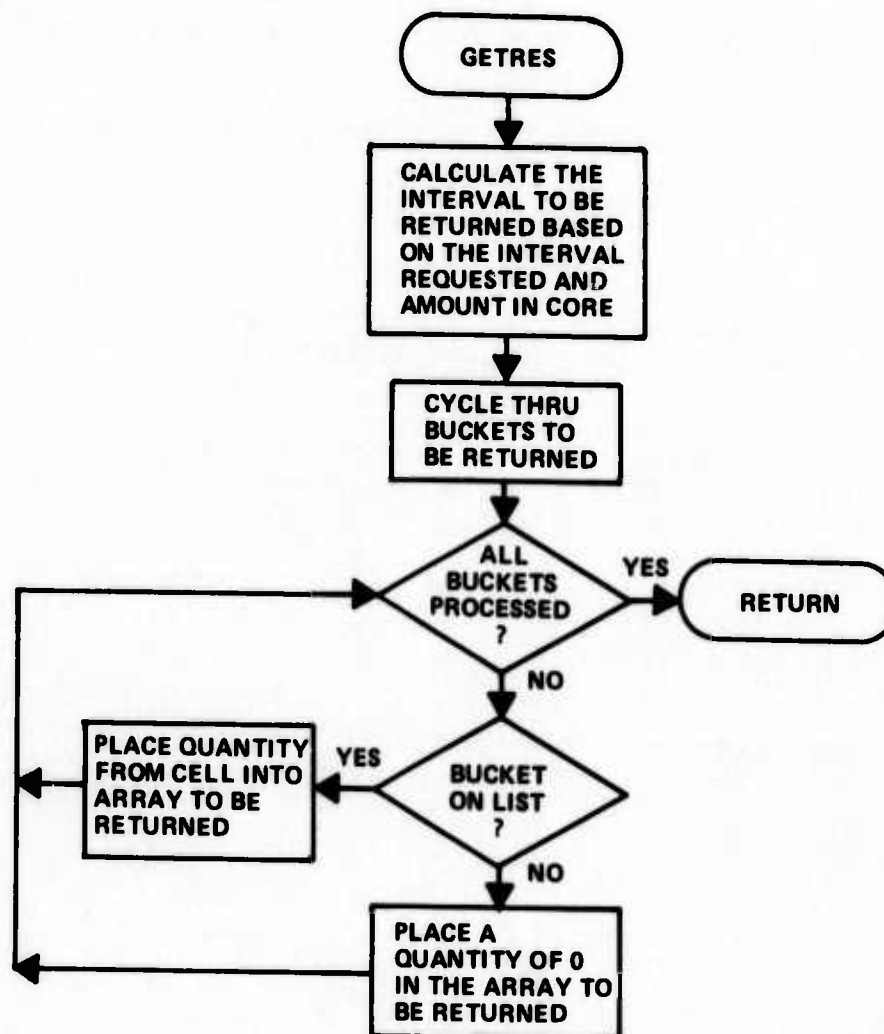


```

C***** GETRES *****
C*
C*          SUBROUTINE GETRES
C*
C*  PURPOSE
C*    READS QUANTITY OF A GIVEN RESOURCE FOR A GIVEN PERIOD.
C*
C*  CALLING SEQUENCE
C*    CALL GETRES(IRES,ITIIN,IT2IN,IT1OUT,IT2OUT,IARRAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    IRES   - RESOURCE NUMBER
C*    IT1IN  - BEGINNING OF TIME INTERVAL REQUESTED
C*    IT2IN  - END OF TIME INTERVAL REQUESTED
C*
C*          * EXPLICIT OUTPUT *
C*    IT1OUT - BEGINNING OF TIME INTERVAL RETURNED
C*    IT2OUT - END OF TIME INTERVAL RETURNED
C*    IARRAY - ARRAY OF QUANTITIES RETURNED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*****

```

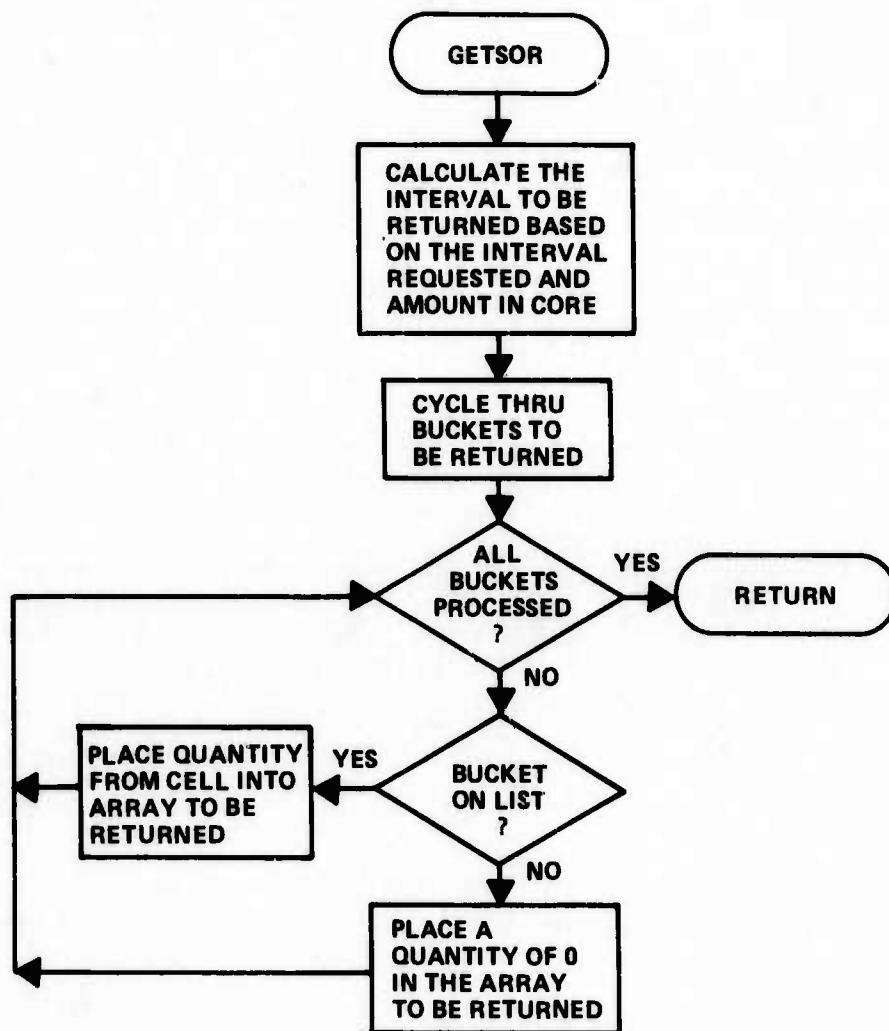




```

C***** GETSOR *****
C*
C*
C*          SUBROUTINE GETSOR
C*
C*  PURPOSE
C*    READS QUANTITY OF A GIVEN SOURCE FOR A GIVEN PERIOD.
C*
C*  CALLING SEQUENCE
C*    CALL GETSOR(ISOR,IT1IN,IT2IN,IT1OUT,IT2OUT,IARRAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    ISOR   - SOURCE NUMBER
C*    IT1IN  - BEGINNING OF TIME INTERVAL REQUESTED
C*    IT2IN  - END OF TIME INTERVAL REQUESTED
C*
C*          * EXPLICIT OUTPUT *
C*    IT1OUT - BEGINNING OF TIME INTERVAL RETURNED
C*    IT2OUT - END OF TIME INTERVAL RETURNED
C*    IARRAY - ARRAY OF QUANTITIES RETURNED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*****

```

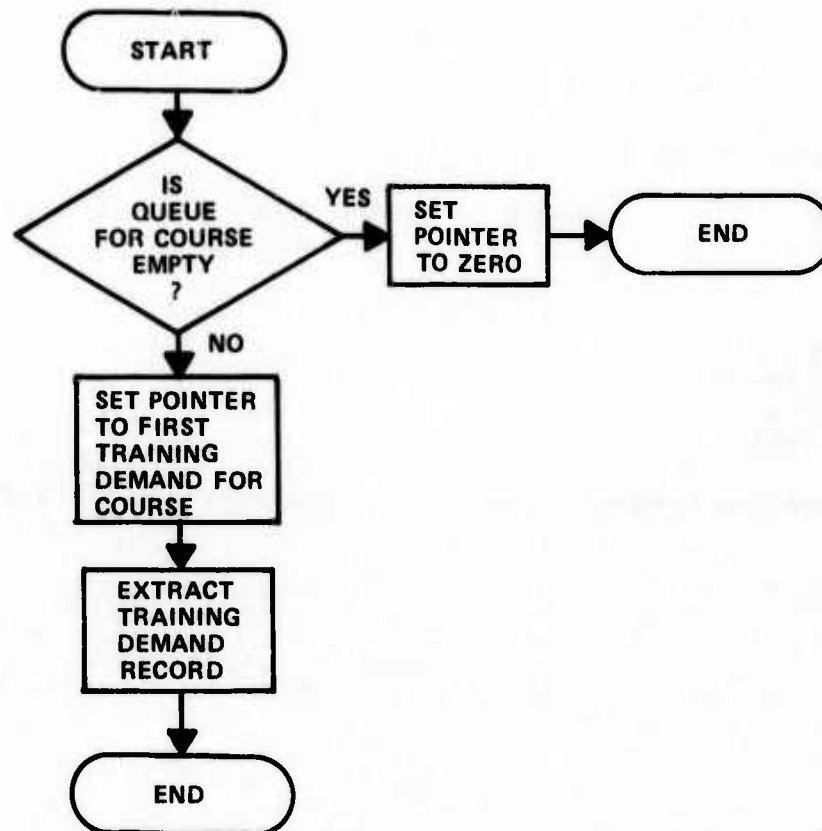


```

CC***** GETTDB *****
CC*
CC*  PURPOSE
CC*    UTILITY ROUTINE TO ACCESS TRAINING DEMAND INFORMATION
CC*    STORED IN A LINKED LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL GETTDB(NCORSE,IPOINT,NUMT,IDATE,IDGRAD,ITTYPE,IDTYPE)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    NCORSE          COURSE NUMBER FOR WHICH NEXT AVAILABLE
CC*                   TRAINING DEMAND IS REQUIRED.
CC*
CC*    * OUTPUT *
CC*
CC*    IPOINT          ADRESS OF NEXT AVAILABLE TRAINING DEMAND
CC*                   RECORD. 0 IF NONE REMAIN.
CC*    NUMT            NUMBER OF STUDENTS.(FLOATING POINT VALUE)
CC*    IDATE           DEMAND DATE
CC*    IDGRAD          GRADUATION ID. (SET TO ZERO IN ADDTDQ)
CC*    ITTYPE          TRAINEE TYPE (PILOT,COPILOT,OSO,DSO).
CC*    IDTYPE          REASON FOR DEMAD. (CCTS DUE TO DELIVERY,
CC*                   CCTS DUE TO ATTRITION,PMT).
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# GETTDB



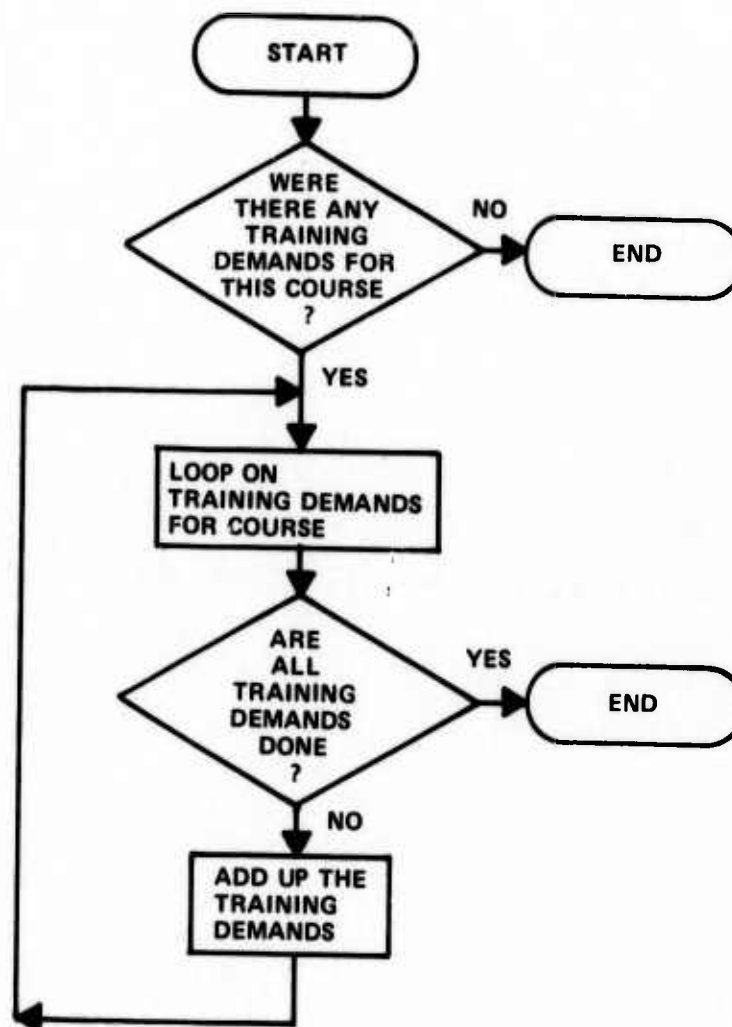


```

CC***** GRADF *****
CC*
CC*  PURPOSE
CC*    TO ACCUMULATE THE TRAINING DEMANDS FOR A COURSE.
CC*
CC*  REMARKS
CC*    THIS ROUTINE IS USED ONLY WHEN CREWS CAN BE FORMED
CC*    WITHOUT REGARD TO THE DESTINATION AIR BASE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL GRADF(NCORSE)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NCORSE          NUMBER OF COURSE FOR WHICH TRAINING DEMAND
CC*                   RECORDS ARE TO BE ACCUMULATED.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# GRADF

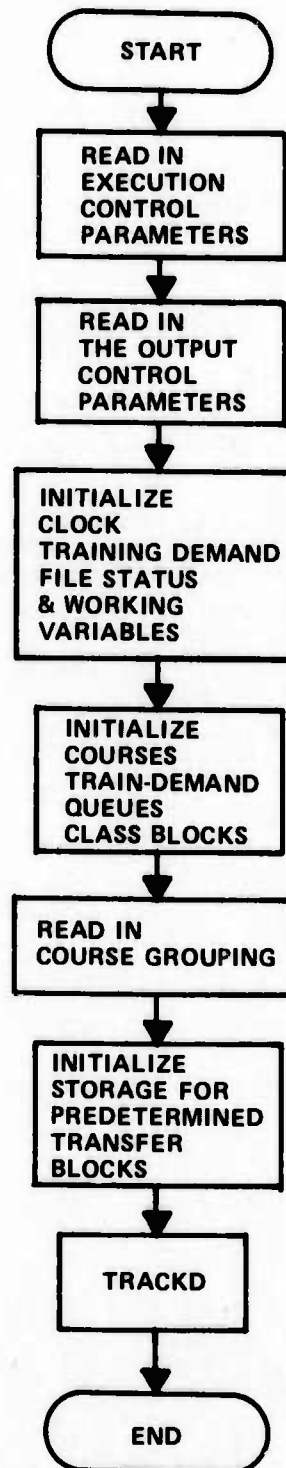


```

CC***** INIT *****
CC*
CC*  PURPOSE
CC*    1. READ IN EXECUTION AND OUTPUT CONTROL PARAMETERS.
CC*    2. INITIALIZE WORKING STORAGE AND POINTERS.
CC*    3. CREATE TRACK DESCRIPTOR RECORDS
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL INIT
CC*
CC*  REMARKS
CC*    THE ROUTINE MUST BE PROCESSED BY THE VARY PROGRAM BEFORE
CC*    COMPILATION BECAUSE IT CONTAINS VARIABLES THAT DEFINE THE
CC*    SIZE OF THE ARRAYS.
CC*
CC*  SUBROUTINES USED
CC*    TRACKD
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

INIT



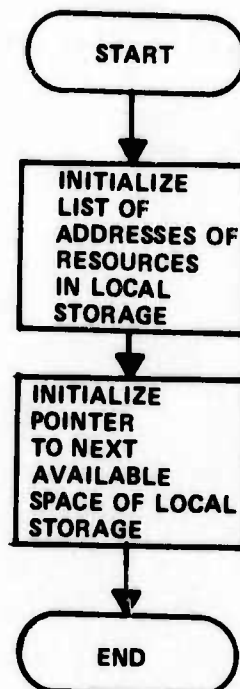
```

CC***** INITR *****
CC*
CC*  PURPOSE
CC*    INITIALIZE WORKING STORAGE FOR TENTATIVE RESOURE ALLOCATION
CC*    CALCULATIONS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL INTR
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



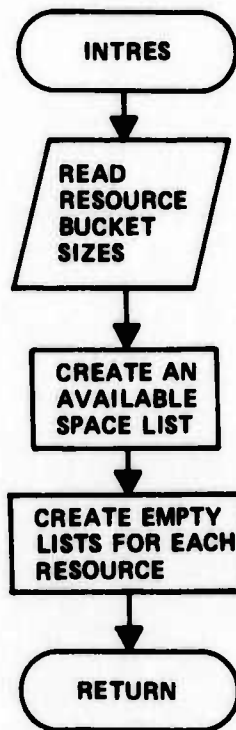
# INTR



```

C***** INTRES *****
C*
C*
C*          SUBROUTINE INTRES
C*
C*  PURPOSE
C*    INITIALIZE RESOURCE TABLES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*
C*****

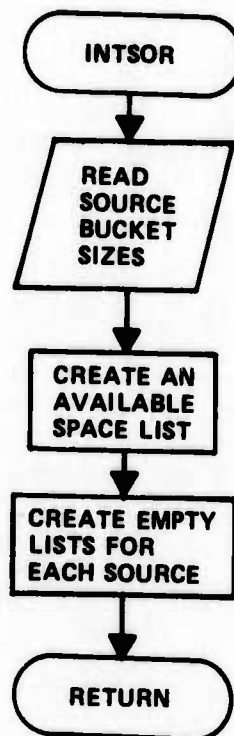
```



```

C***** INTSOR *****
C*
C*          SUBROUTINE INTSOR
C*
C*  PURPOSE
C*    INITIALIZE SOURCE TABLES
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    26 APRIL 1975
C*
C*****

```



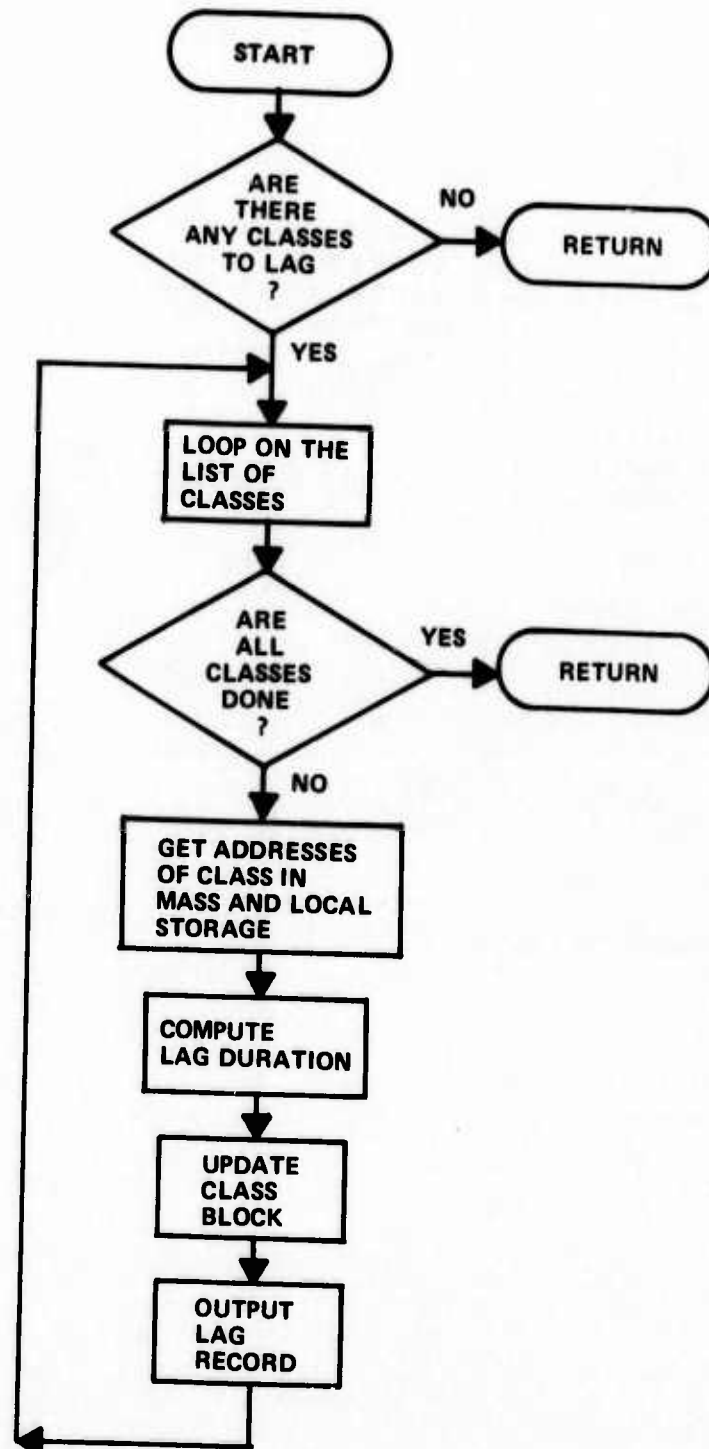
```

CC***** LAG *****
CC*
CC*  PURPOSE
CC*    TO DELAY PROCESSING OF A SET OF CLASSES UNTIL A SPECIFIED
CC*    TIME.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL LAG(NOCLS,INDXC,ITIME,ITFNCT,ITASKA,IPBDUR,IDFRES)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NOCLS      NUMBER OF CLASSES TO BE LAGGED.
CC*    INDXC()    LIST OF CLASSES TO BE LAGGED.
CC*    ITIME      TIME TO WHICH CLASSES WILL BE LAGGED
CC*    ITFNCT     FUNCTION OF THE TASK THAT CAUSED THE LAG.
CC*    ITASKA     ADDRESS OF THE TASK THAT CAUSED THE LAG.
CC*    IPBDUR     DURATION OF THE PROCBLOC.
CC*    IDFRES     NUMBER OF THE RESOURCE CAUSING THE LAG.
CC*               (0 IF LAG IS DUE TO SYNC OR CORR)
CC*
CC*  SUBROUTINES USED
CC*    BLOCK
CC*
CC*  PROGRAMMER
CC*    G.GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



# LAG

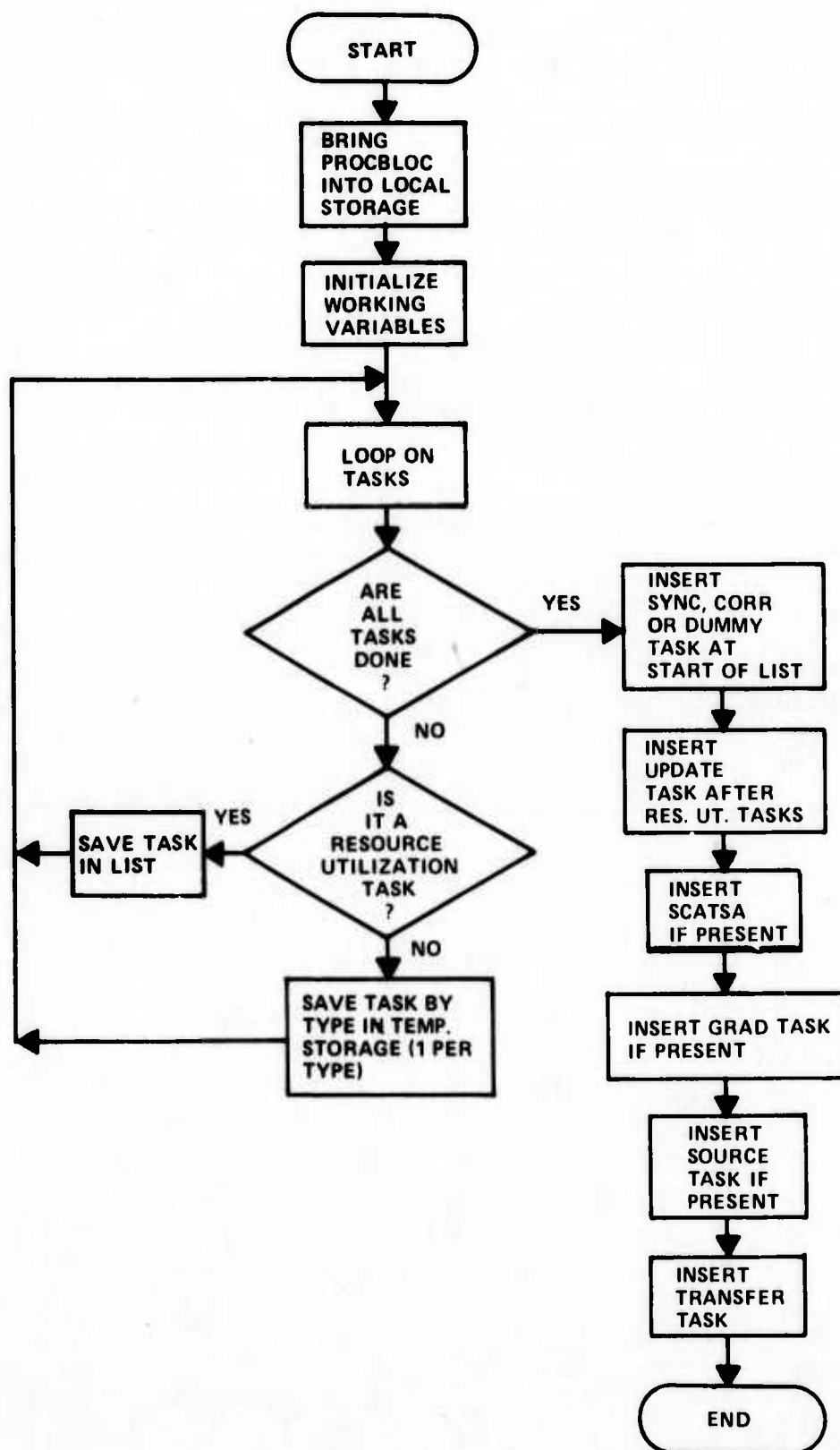


```

CC***** LSTASK *****
CC*
CC*  PURPOSE
CC*    TO CREATE A SEQUENTIAL LIST OF THE TASKS IN A PROCBLOC.
CC*    SYNCHRONIZING TASKS ARE PLACED AT THE BEGINNING OF THE
CC*    LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL LSTASK
CC*
CC*  REMARKS
CC*    ONLY 1 OF EACH TYPE OF NON-RESOURCE UTILIZATION TASKS
CC*    ARE USED. THE TASKS ARE ARRANGED IN THE FOLLOWING ORDER:
CC*      1. SYNCHRONIZATION OR CORRELATION TASKS.
CC*      2. RESOURCE UTILIZATION TASKS.
CC*      3. UPDATE TASK. (PROVIDED BY PROGRAM).
CC*      4. SOURCE ALLOCATION. (SCATSA)
CC*      5. GRADUATION.
CC*      6. GET SOURCE TASK.
CC*      7. TRANSFER TASK. (PROVIDED BY PROGRAM).
CC*
CC*  SUBROUTINES USED
CC*    BLOCK
CC*    PLIST
CC*    PBLOCK
CC*    CBLOCK
CC*    TBLOCK
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# LSTASK

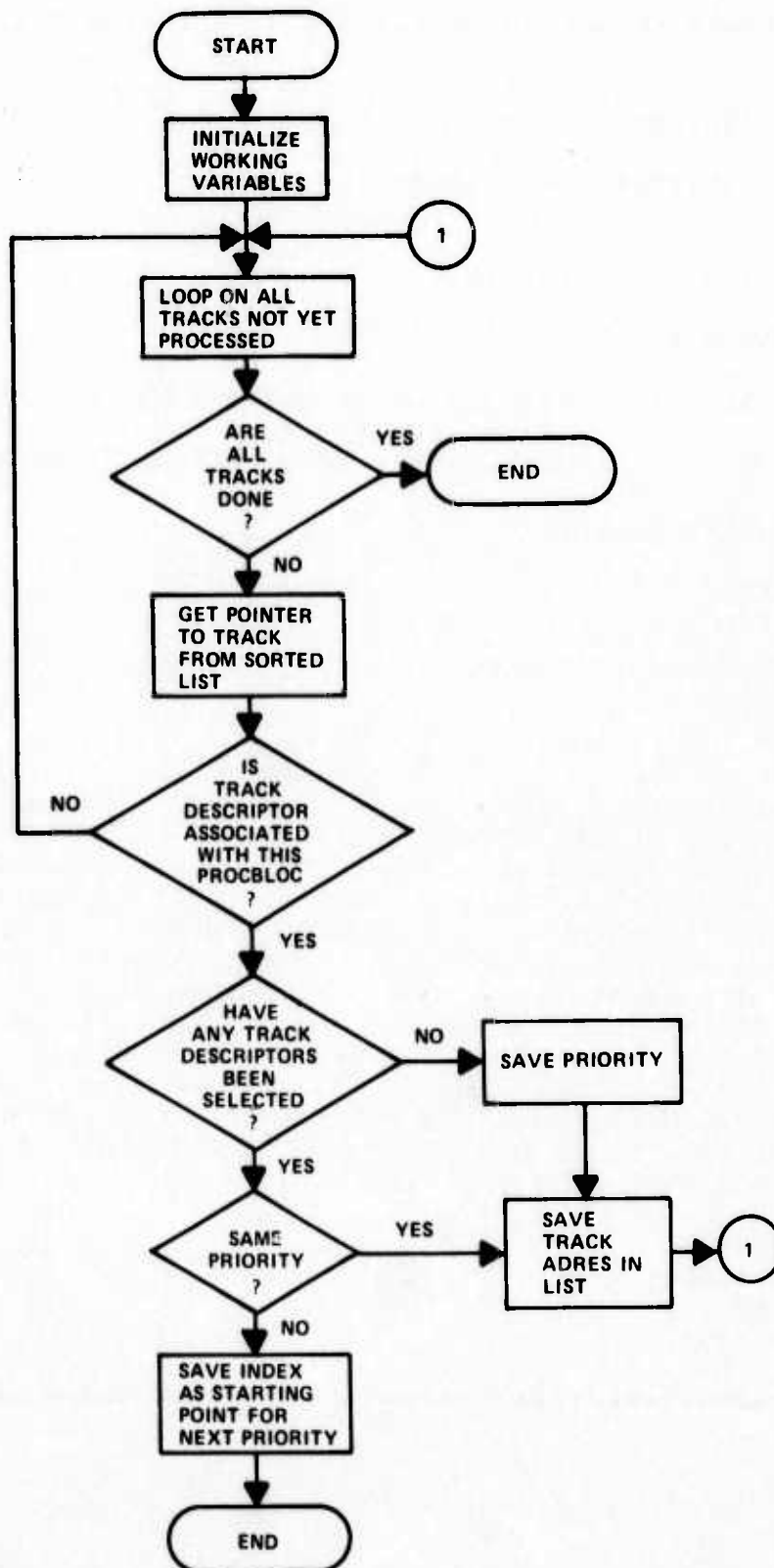


```

CC***** LSTRAK *****
CC*
CC*  PURPOSE
CC*    GET A LIST OF TRACK DESCRIPTOR BLOCKS WITH EQUAL PRIORITIES
CC*    AND ASSOCIATED WITH THE SPECIFIED PROCBLOC.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL LSTRAK(L1,L2,NPROCB,  N,LIST )
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    L1          POINTER TO INDEX OF FIRST TDB FOR COURSE.
CC*    L2          POINTER TO INDEX OF LAST TDB FOR COURSE.
CC*    NPROCB      ADRESS OF PROCBLOC THAT THE CLASS IS IN.
CC*
CC*    * OUTPUT *
CC*
CC*    N           NUMBER OF ELEMENTS IN LIST
CC*    LIST()      INDICES OF TRACK DESCRIPTOR BLOCKS THAT
CC*                HAVE EQUAL PRIORITIES.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

# LSTRAK

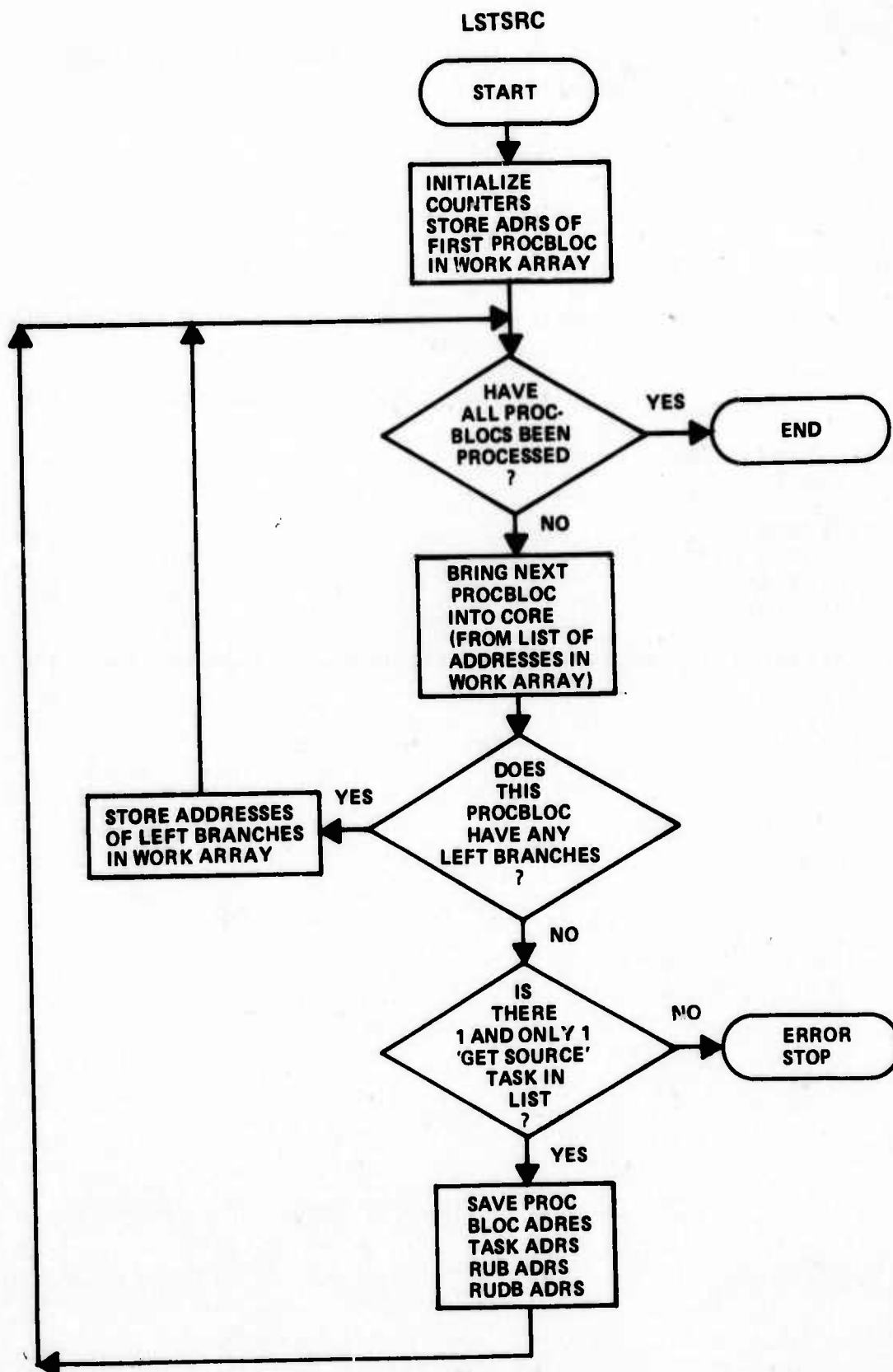


```

CC***** LSTSRC *****
CC*
CC*  PURPOSE
CC*    TO GENERATE A LIST OF ALL THE SOURCES FOR A COURSE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SRTSRC(IADPB1,IDUMP )
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IADPB1      ADRESS OF GRADUATION BLOCK FOR COURSE
CC*                BEING PROCESSED.
CC*    IDUMP       DIAGNOSTIC DUMP SWITCH. (1-PRINT DIAGS.)
CC*
CC*  SUBROUTINES CALLED
CC*
CC*    BLOCK
CC*    PBLOCK
CC*    TBLOCK
CC*    WRUB
CC*    WRUDB
CC*
CC*  REMARKS
CC*    THIS ROUTINE STARTS AT THE RIGHTMOST (GRADUATION) PROCBLOC
CC*    OF A COURSE AND STEPS BACK THRU THE PROCBLOCKS (USING THE
CC*    BACKWARD POINTERS) UNTIL A PROCBLOC IS REACHED THAT DOES
CC*    NOT HAVE ANY BACKWARD POINTERS. THIS PROCBLOC IS ASSUMED
CC*    TO BE THE SOURCE BLOCK AND IS CHECKED TO MAKE SURE IT HAS
CC*    ONE AND ONLY ONE GETSOURCE TASK ASSOCIATED WITH IT. A
CC*    FURTHER CHECK IS THEN MADE TO ASSURE THAT ONLY ONE RUDB
CC*    IS DEFINED FOR THE SOURCE AND HAS NO ALTERNATES OR SECONDARY
CC*    RUBS. IF ALL CONDITIONS ARE SATISFIED, THE ADRESSES OF THE
CC*    SOURCE PROCBLOC,TASK,RUB AND RUDB AND THE SOURCE NUMBER
CC*    ARE SAVED IN ARRAYS IN THE COMMON BLOCK SORDSC.
CC*    THIS PROCESS IS REPEATED UNTIL ALL THE SOURCES IN THE
CC*    COURSE HAVE BEEN IDENTIFIED.
CC*
CC*  PROGRAMMER
CC*    G.GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```



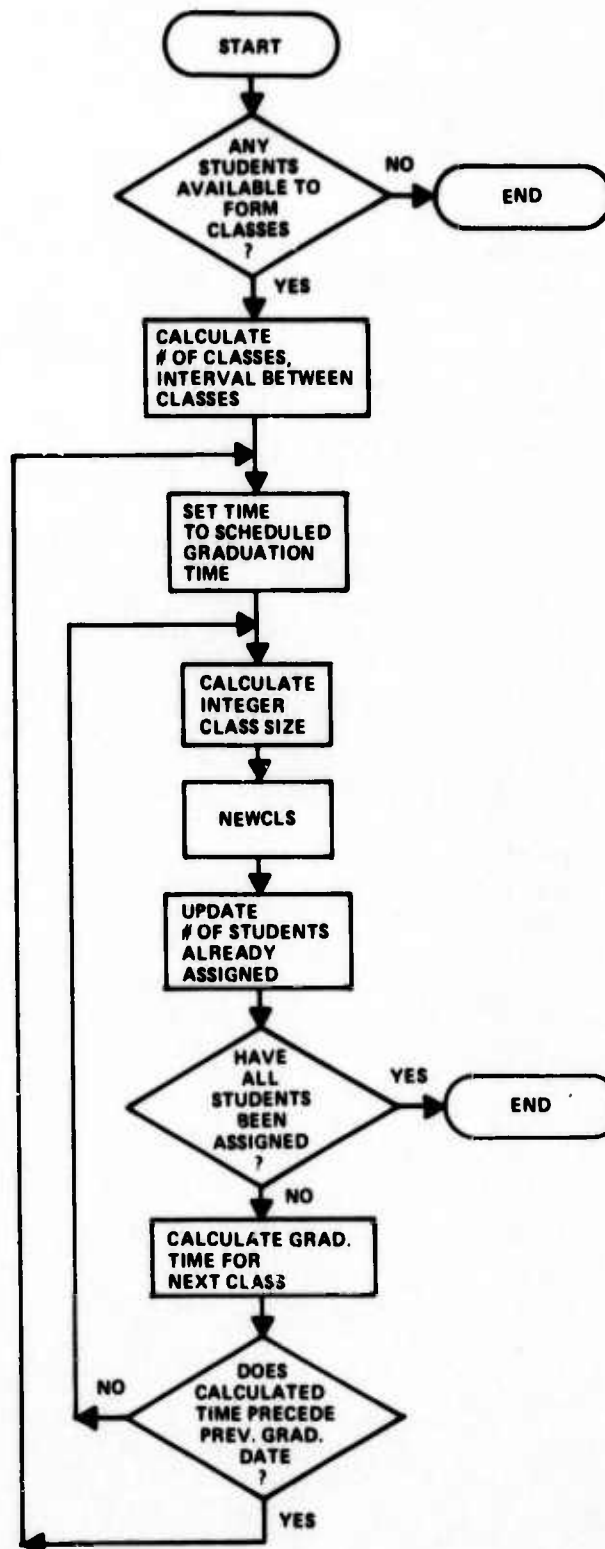


```

CC***** MLTCLS *****
CC*
CC*  PURPOSE
CC*    FORM MULTIPLE CLASSES FROM THE ACCUMULATED TRAINING
CC*    DEMANDS ON A COURSE
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL MLTCLS(ITOTD,NCORSE,IGRID)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    ITOTD          NUMBER OF STUDENTS TO BE ASSIGNED TO CLASSES.
CC*    NCORSE        COURSE NUMBER
CC*    IGRID         GRADUATION ID. COUNTER.
CC*                  (IF IGRID=-1, CLASS IS AN EXTRAS CLASS)
CC*
CC*  SUBROUTINES USED
CC*    NEWCLS
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

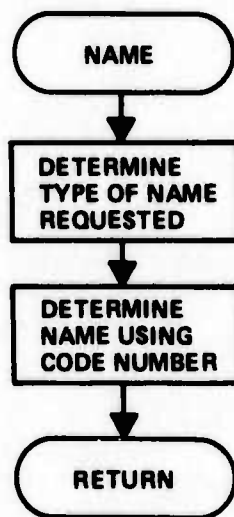
# MLTCLS



```

C***** NAME *****
C*
C* SUBROUTINE NAME
C*
C* PURPOSE
C* RETURN A NAME FOR CODE NUMBER.
C*
C* CALLING SEQUENCE
C* CALL NAME(IAPRV,NUMBER,INAME)
C*
C* DESCRIPTION OF PARAMETERS
C*
C* * EXPLICIT INPUT *
C* IAPRV - ALPHANUMERIC NAME OF THE TYPE OF NAME BEING LOOKED UP*
C* 'AB' - AIR BASE NAME
C* 'C' - COURSE NAME
C* 'GF' - GRADUATION FUNCTION NAME
C* 'PB' - PROC BLOCK NAME
C* 'PBNU' - PROC BLOCK NUMBER
C* 'R' - RESOURCE NAME
C* 'RUB' - RUB NAME
C* 'RUDB' - RUDB NAME
C* 'RUGF' - RESOURCE UTILIZATION FUNCTION NAME
C* 'RUTF' - RESOURCE UTILIZATION TIMING FUNCTION
C* 'S' - SOURCE NAME
C* 'TB' - TASK BLOCK NAME
C* 'TF' - TASK FUNCTION NAME
C* NUMBER - CODE NUMBER OF NAME BEING LOOKED UP
C*
C* * EXPLICIT OUTPUT *
C* INAME - ALPHANUMERIC NAME BEING RETURNED
C*
C* AUTHOR/PROGRAMMER
C* JOHN R. MENIG
C* CALSPAN CORPORATION
C* 22 APRIL 1975
C*
C*****

```



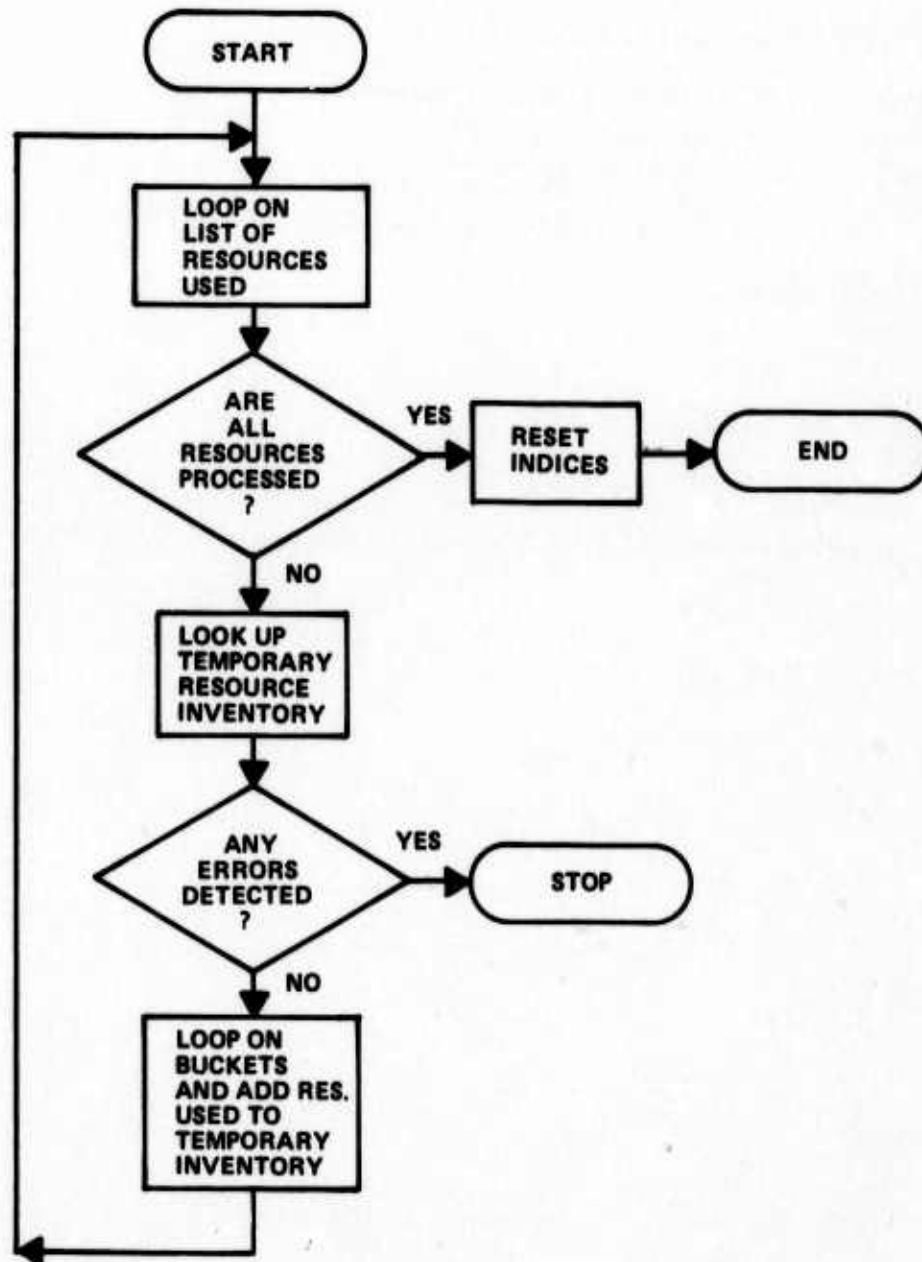
```

CC***** NEGUSE *****
CC*
CC*  PURPOSE
CC*    TO NEGATE TEMPORARY UPDATES OF RESOURCE INVENTORIES.
CC*    (WHEN A PRIMARY IS SATISFIED BUT A SECONDARY CANNOT
CC*    BE SATISFIED).
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL NEGUSE
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    AS DESCRIBED IN SVRUS1 AND SVRUS2
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```



# NEGUSE

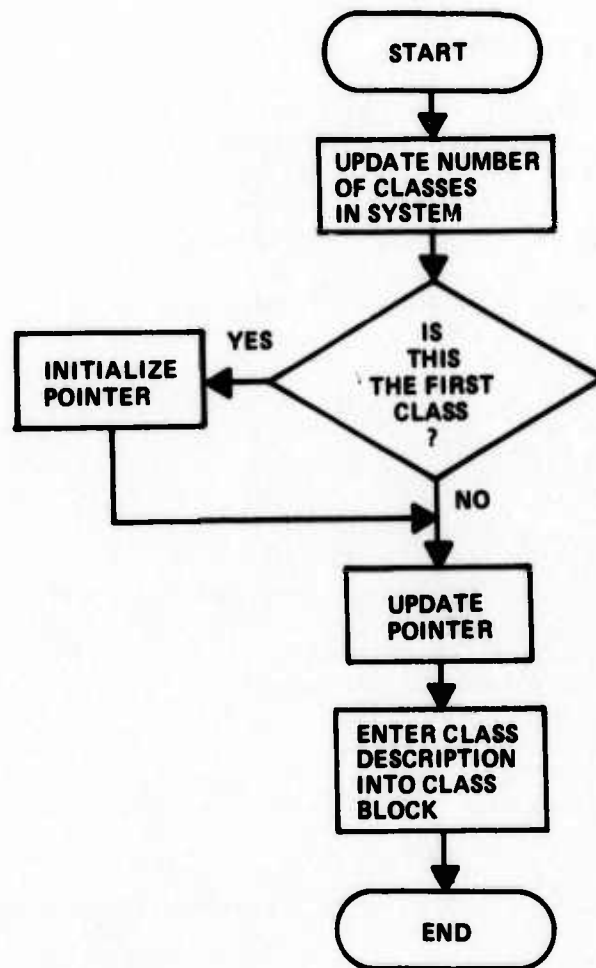


```

CC***** NEWCLS *****
CC*
CC*  PURPOSE
CC*    TO GENERATE A CLASS BLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL NEWCLS(NSTDS,NCORSE,IGDTE,IGRID)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NSTDS          NUMBER OF STUDENTS IN CLASS.
CC*    NCORSE         COURSE NUMBER
CC*    IGDTE          GRADUATION DATE
CC*    IGRID          GRADUATION ID COUNTER
CC*                  (IF IGRID=-1,CLASS IS AN EXTRAS CLASS)
CC*
CC*  SUBROUTINES USED
CC*    CLSDMP
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

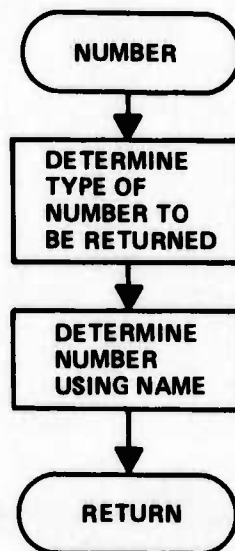
# NEWCLS



```

C***** NUMBER *****
C*
C*
C*          SUBROUTINE NUMBER
C*
C* PURPOSE
C*   RETURNS A CODE NUMBER FOR A NAME.
C*
C* CALLING SEQUENCE
C*   CALL NUMBER(IAPRV,NUMB,NAME)
C*
C* DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*   IAPRV - ALPHANUMERIC NAME OF THE TYPE OF CODE BEING LOOKED UP
C*           'AB' - AIR BASE NAME
C*           'C' - COURSE NAME
C*           'GF' - GRADUATION FUNCTION NAME
C*           'PB' - PROC BLOCK NAME
C*           'PBNU' - PROC BLOCK NUMBER
C*           'R' - RESOURCE NAME
C*           'RUB' - RUB NAME
C*           'RUDB' - RUDB NAME
C*           'RUGF' - RESOURCE UTILIZATION FUNCTION NAME
C*           'RUTF' - RESOURCE UTILIZATION TIMING FUNCTION
C*           'S' - SOURCE NAME
C*           'TB' - TASK BLOCK NAME
C*           'TF' - TASK FUNCTION NAME
C*   NAME - ALPHANUMERIC NAME BEING LOOKED UP
C*
C*          * EXPLICIT OUTPUT *
C*   NUMB - CODE NUMBER RETURNED
C*
C* AUTHOR/PROGRAMMER
C*   JOHN R. MENIG
C*   CALSPAN CORPORATION
C*   22 APRIL 1975
C*
C*****

```



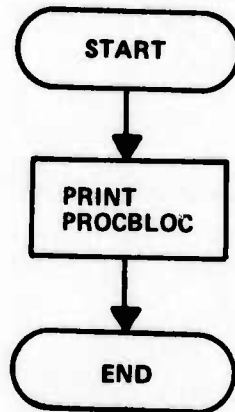
```

CC***** PBLOCK *****
CC*
CC*  PURPOSE
CC*    TO PRINT A PROCBLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL PBLOCK(IADRS,IBLOCK)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS      ADRESS OF PROCBLOCK
CC*    IBLOCK     FIRST WORD OF PROCBLOC
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



PBLOCK

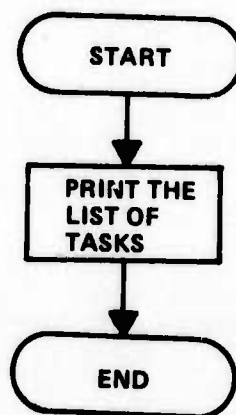


```

CC***** PLIST *****
CC*
CC*  PURPOSE
CC*    TO PRINT THE LIST OF TASKS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL PLIST
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# PLIST

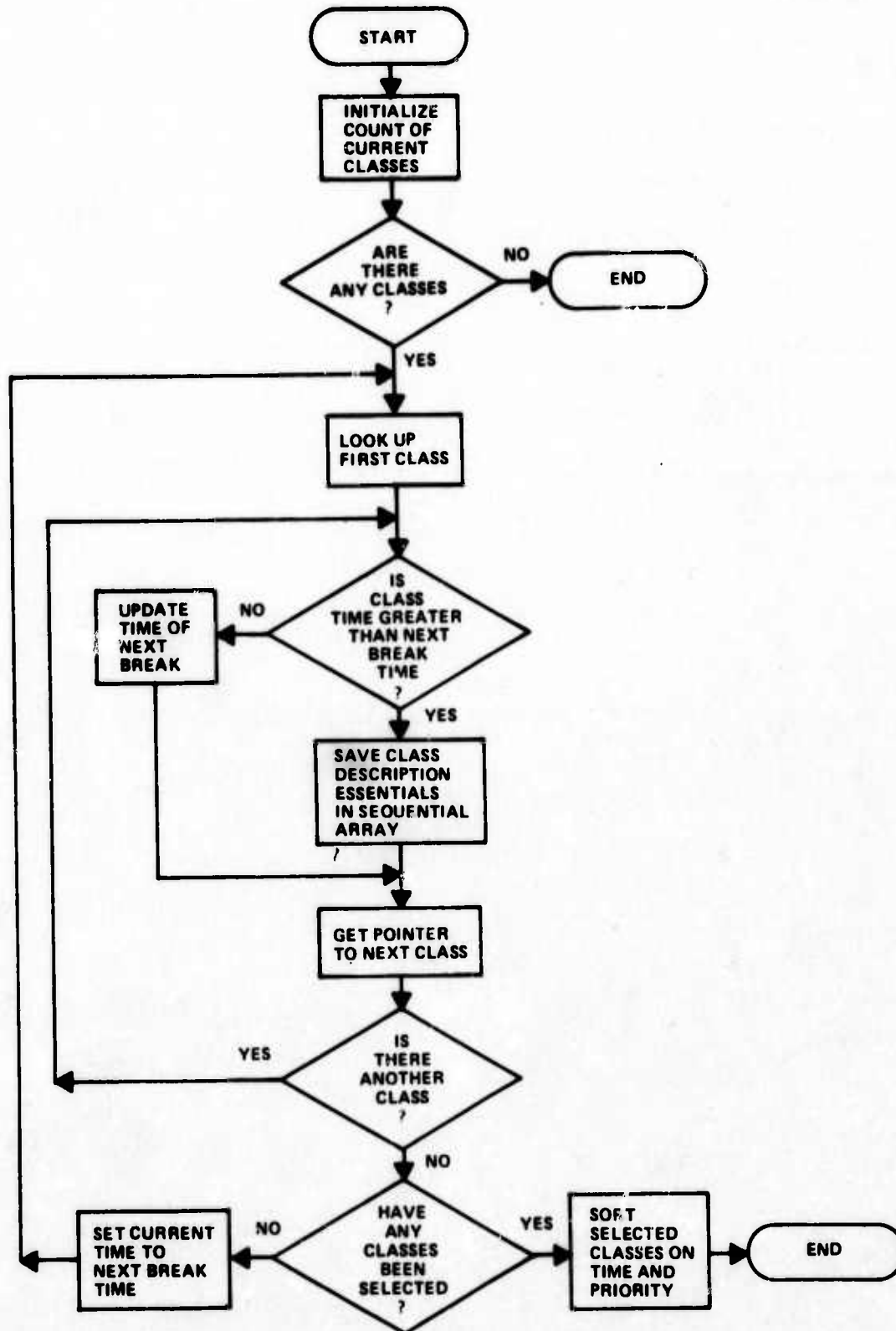


```

CC***** PREPC *****
CC*
CC*  PURPOSE
CC*    SELECT CURRENTLY ACTIVE CLASSES AND SORT IN ORDER BY
CC*    TIME AND PRIORITY.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL PREPC
CC*
CC*  SUBROUTINES USED
CC*    CLSDMP
CC*    SRTCTP
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

PREPC



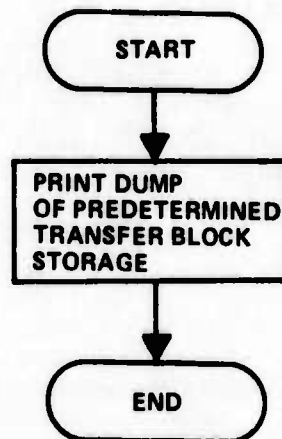
```

CC***** PTBDMP *****
CC*
CC*  PURPOSE
CC*    TO PRINT A DUMP OF THE PREDETERMINED TRANSFER BLOCKS
CC*    STORAGE.
CC*
CC*  CALLING SEQUENCE
CC*    CALL PTBDMP(IADRS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS          ADDRESS OF BAD PTB
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



PTBDMP

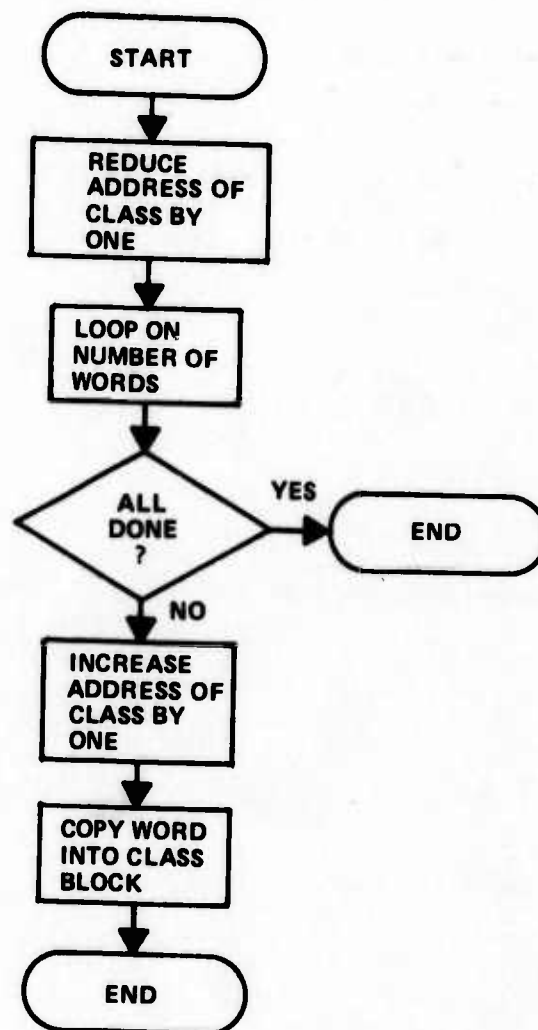


```

CC***** PUTCLS *****
CC*
CC*  PURPOSE
CC*    TO REPLACE THE CONTENT OF A CLASS BLOCK IN LINKED STORAGE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL PUTCLS(INDEX,IA,N)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    INDEX      ADRESS OF CLASS BLOCK IN MASS STORAGE.
CC*    IA()      LOCAL STORAGE FOR CLASS BLOCK.
CC*    N         NUMBER OF WORDS IN PROCBLOC TO BE WRITTEN
CC*              IN MASS STORAGE.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# PUTCLS

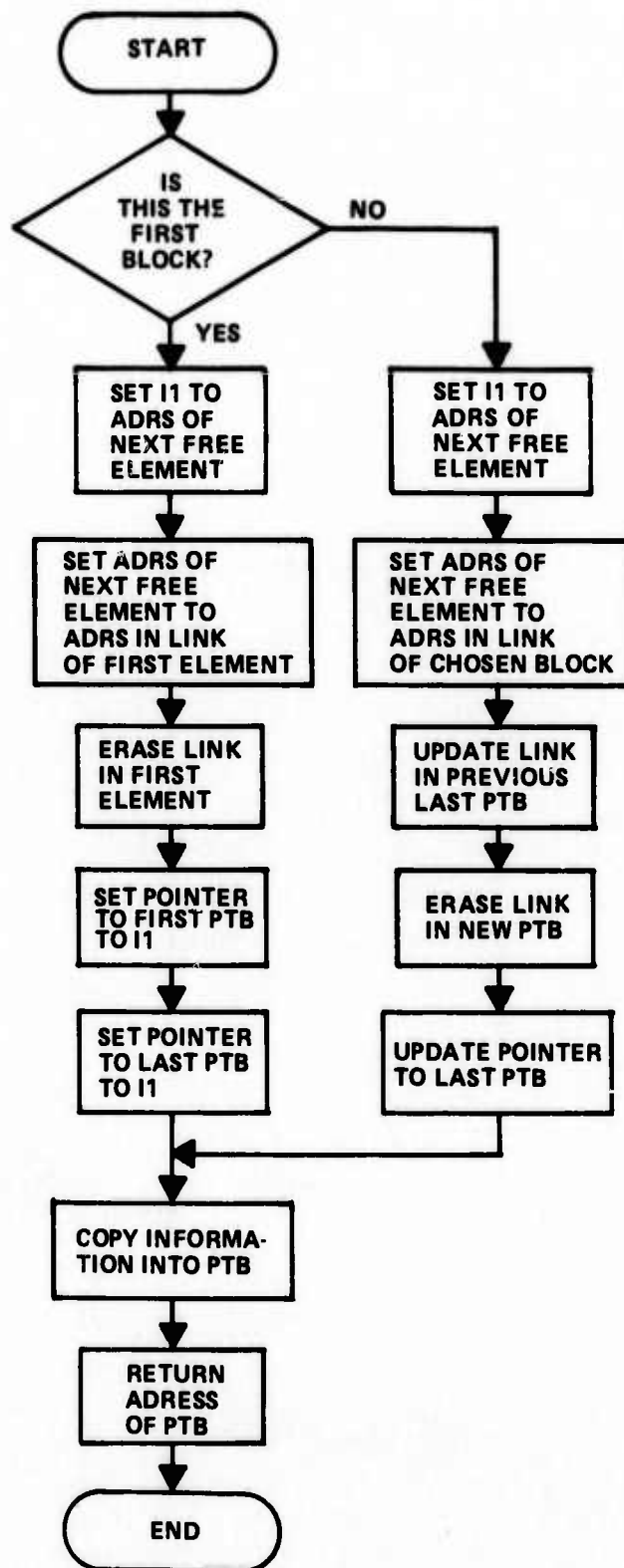


```

CC***** PUTPTB *****
CC*
CC*  PURPOSE
CC*    TO CREATE A PREDETERMINED TRANSFER BLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL PUTPTB(IPROP,NXT,IBLKN)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IPROP()      NUMBER OF STUDENTS TO BE SENT ALONG EACH ONE
CC*                OF THE 5 BRANCHES OF THE PROCBLOC.
CC*    NXT()        ADRESS OF THE NEXT PTB ALONG THE TRACK.
CC*
CC*    * OUTPUT *
CC*
CC*    IBLKN        ADRESS WHERE PTB WAS STORED.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# PUTPTB

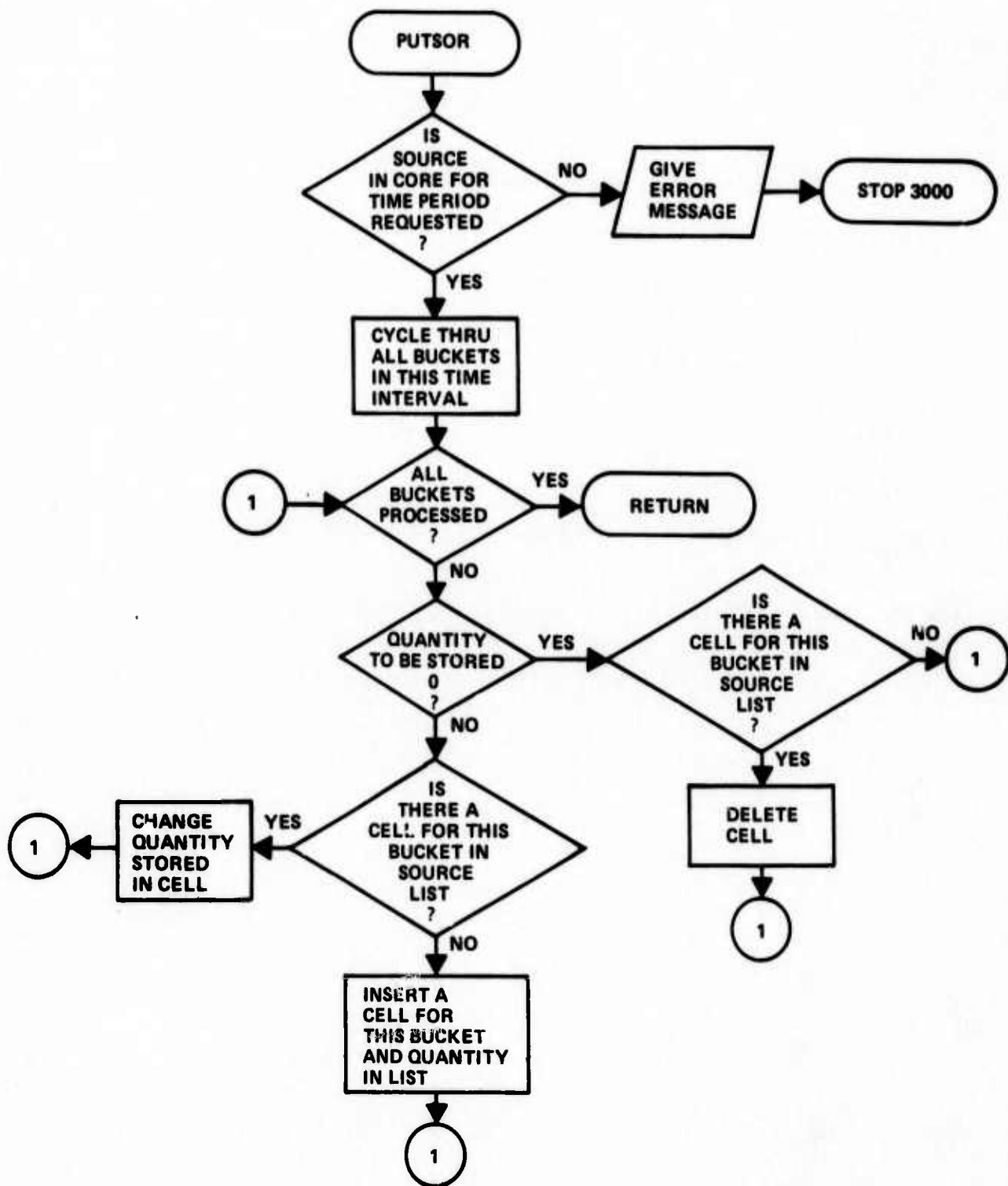


```

C***** PUTSOR *****
C*
C*          SUBROUTINE PUTSOR
C*
C*  PURPOSE
C*    WRITES QUANTITY OF A GIVEN SOURCE FOR A GIVEN PERIOD.
C*
C*  CALLING SEQUENCE
C*    CALL PUTSOR(ISOR,IT1,IT2,IAPPAY)
C*
C*  DESCRIPTION OF PARAMETERS
C*
C*          * EXPLICIT INPUT *
C*    ISOR   - SOURCE NUMBER
C*    IT1    - BEGINNING OF INTERVAL
C*    IT2    - END OF INTERVAL
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    29 APRIL 1975
C*****

```

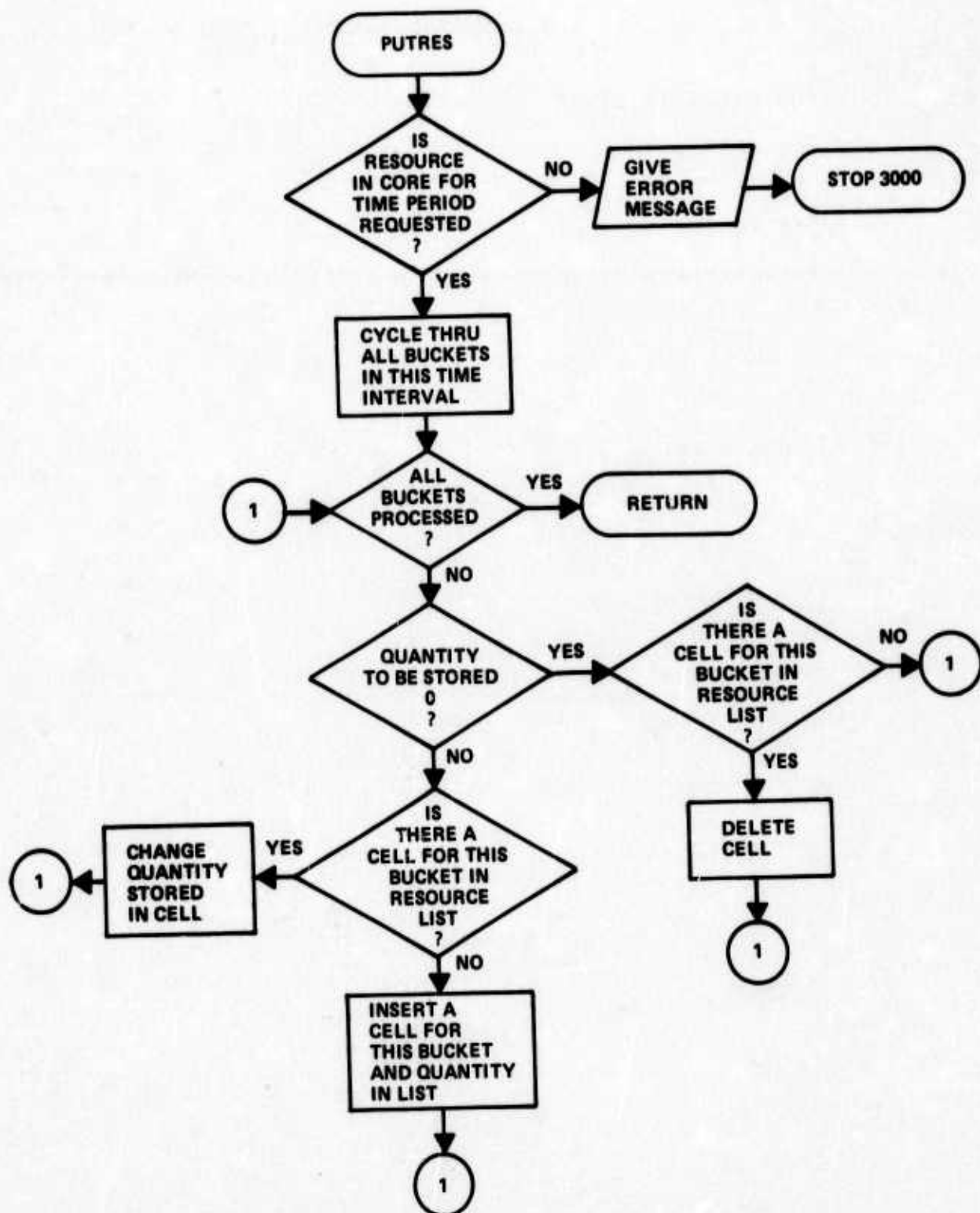




```

C***** PUTRES *****
C*
C* SUBROUTINE PUTRES *
C*
C* PURPOSE *
C* WRITES QUANTITY OF A GIVEN RESOURCE FOR A GIVEN PERIOD *
C*
C* CALLING SEQUENCE *
C* CALL PUTRES(IRES,IT1,IT2,IARRAY) *
C*
C* DESCRIPTION OF PARAMETERS *
C*
C* * EXPLICIT INPUT *
C* IRES - RESOURCE NUMBER *
C* IT1 - BEGINNING OF INTERVAL *
C* IT2 - END OF INTERVAL *
C*
C* AUTHOR/PROGRAMMER *
C* JOHN R. MENIC *
C* CALSPAN CORPORATION *
C* 29 APRIL 1975 *
C*****

```



```

C***** RDNAME *****
C*
C*
C*          SUBROUTINE RDNAME
C*
C*  PURPOSE
C*    READS NAMES IN STEP3
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    22 APRIL 1975
C*
C*****

```



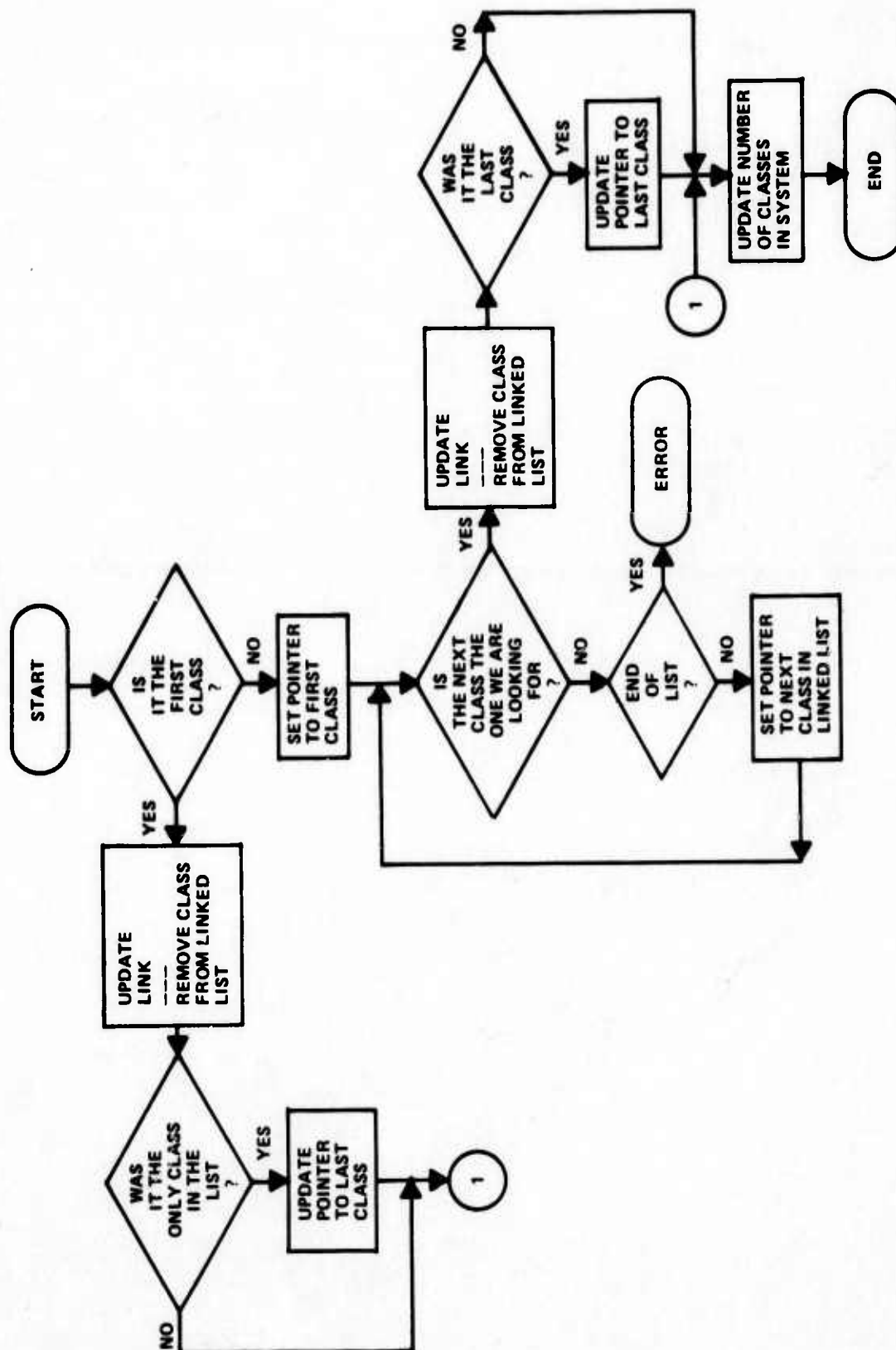
```

CC***** REMCLS *****
CC*
CC*  PURPOSE
CC*    TO FREE THE STORAGE SPACE OCCUPIED BY A CLASS BLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL REMCLS(IADRS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS          ADRESS OF CLASS TO BE REMOVED FROM STORAGE.
CC*
CC*  SUBROUTINES USED
CC*    CLSDMP
CC*
CC*  PROGRAMMER
CC*    G.GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```



# REMCLS

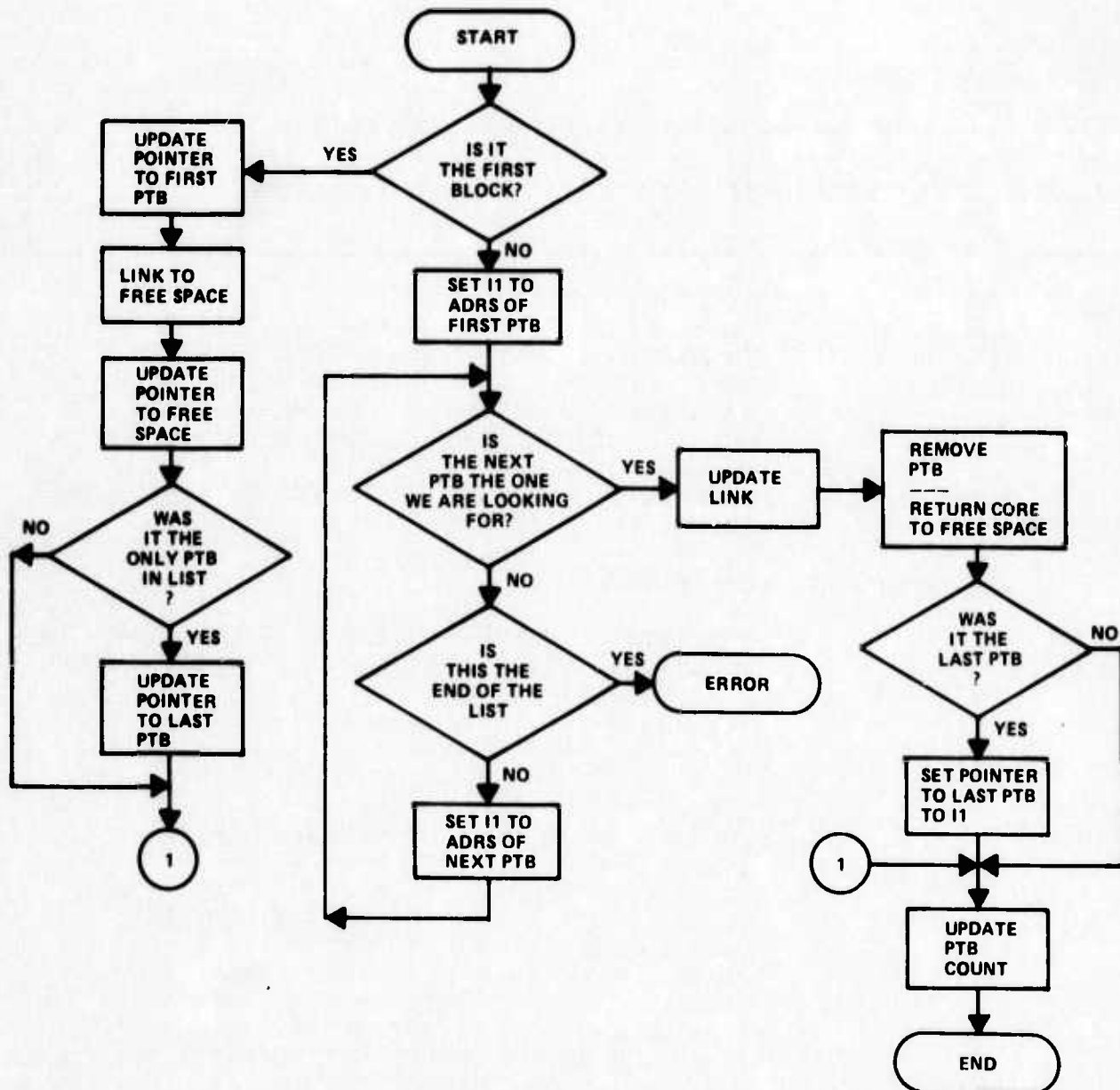


```

CC***** REMPTB *****
CC*
CC* PURPOSE
CC* TO FREE THE STORAGE OCCUPIED BY A PREDETERMINED TRANSFER
CC* BLOCK.
CC*
CC* CALLING SEQUENCE
CC*
CC* CALL REMPTB(IADRS)
CC*
CC* DESCRIPTION OF PARAMETERS
CC*
CC* IADRS ADDRESS OF PTB TO BE REMOVED FROM STORAGE.
CC*
CC* SUBROUTINES USED
CC* PTBDMP
CC*
CC* PROGRAMMER
CC* G. GAIDASZ
CC* CALSPAN
CC* MAY 1975
CC*
CC*****

```

# REMPTB



```

CC***** RESINV *****
CC*
CC*   PURPOSE
CC*     BRING RESOURCE INVENTORY INTO LOCAL STORAGE.
CC*
CC*   CALLING SEQUENCE
CC*
CC*     CALL RESINV(IRESNO,LITIM1,LITIM2,ITIME1,ITIME2)
CC*
CC*   DESCRIPTION OF PARAMETERS
CC*
CC*     * INPUT *
CC*
CC*     IRESNO      RESOURCE NUMBER.
CC*     LITIM1      LOWER LIMIT OF TIME FOR WHICH THIS RESOURCE
CC*                  IS NEEDED BY ANY OF THE CURRENT TASKS.
CC*     LITIM2      UPPER LIMIT OF TIME FOR WHICH THIS RESOURCE
CC*                  IS NEEDED BY ANY OF THE CURRENT TASKS.
CC*     ITIME1      LOWER LIMIT OF TIME FOR WHICH THIS RESOURCE
CC*                  IS NEEDED BY ACTIVE TASK
CC*     ITIME2      UPPER LIMIT OF TIME FOR WHICH THIS RESOURCE
CC*                  IS NEEDED BY ACTIVE TASK.
CC*
CC*     * OUTPUTS VIA COMMON RES *
CC*
CC*     IT1()       THEORETICAL BUCKET NUMBER OF BUCKET CORRES-
CC*                  PONDING TO LITIM1.(ASSUMING RES. INVENTORY
CC*                  STARTS AT TIME=1).
CC*     IT2()       THEORETICAL BUCKET NUMBER OF BUCKET
CC*                  CORRESPONDING TO LITIM2.
CC*     IA1         THEORETICAL BUCKET NUMBER OF BUCKET
CC*                  CORRESPONDING TO ITIME1.
CC*     IA2         THEORETICAL BUCKET NUMBER OF BUCKET
CC*                  CORRESPONDING TO ITIME2.
CC*     INDX1       POINTER TO ELEMENT IN ARRAY INVRES THAT
CC*                  CORRESPONDS TO THE 'HIGH-TIME' BUCKET
CC*                  OF THE DESIRED RES. INVENTORY.
CC*     INDX2       POINTER TO ELEMENT IN ARRAY INVRES THAT
CC*                  CORRESPONDS TO THE 'LOW-TIME' BUCKET
CC*                  OF THE DESIRED RES.INVENTORY.
CC*     NB1         NUMBER OF RESOURCE BUCKETS REQUIRED TO
CC*                  COVER THE ACTIVE PROCBLOC.
CC*
CC*   SUBROUTINES USED
CC*     GETRES
CC*
CC***

```

CONTINUED

\*\*\*

\*\*\*

✱

✱

\*

\*

✱

✱

**\***

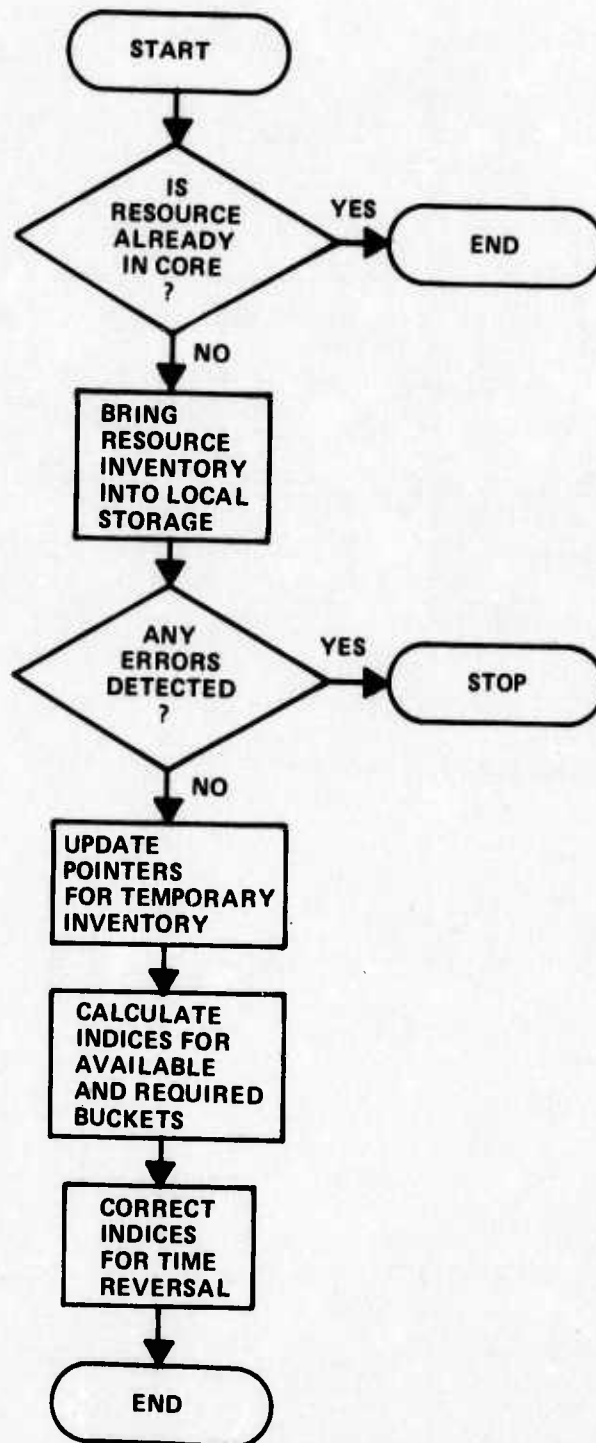
✱

✱



✱

# RESINV



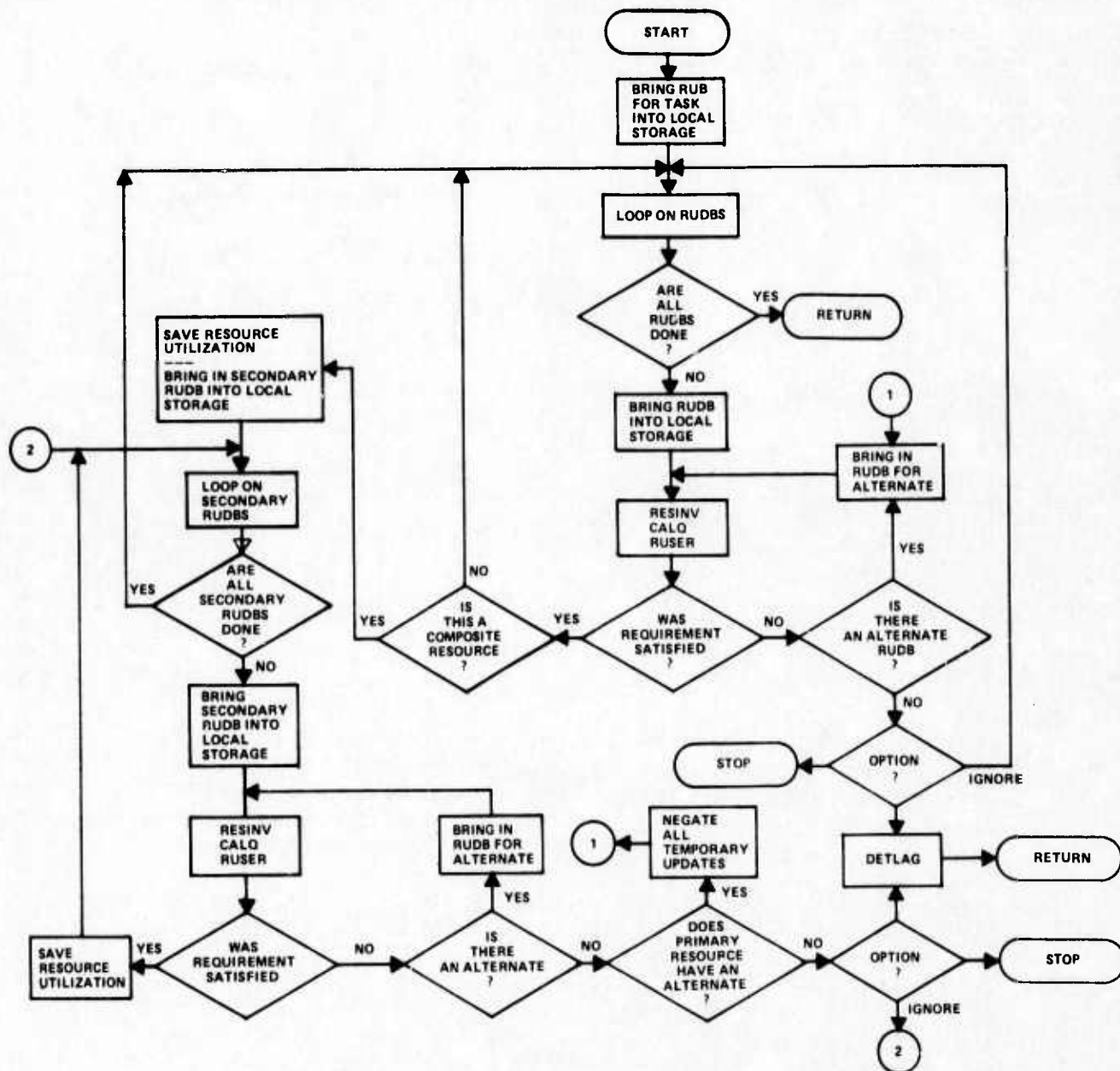


```

CC***** RESUSE *****
CC*
CC* PURPOSE
CC*   TO CALCULATE THE RESOURCES USED BY A CLASS PERFORMING A
CC*   TASK.
CC*
CC* CALLING SEQUENCE
CC*   CALL RESUSE(IFRES,LAGTME)
CC*
CC* DESCRIPTION OF PARAMETERS
CC*
CC*   * OUTPUTS *
CC*   IFRES           NUMBER OF RESOURCE CAUSING ALLOCATION
CC*                   FAILURE.
CC*   LAGTME          TIME TO WHICH CLASS(ES) MUST BE LAGGED.
CC*
CC*   THE ABOVE PARAMETERS HAVE MEANING ONLY IF 'IFATL'
CC*   IN COMMON ECB IS NOT ZERO
CC*
CC* SUBROUTINES CALLED
CC*   RESINV
CC*   CALQ
CC*   RUSER
CC*   SVRUSE
CC*   NEGUSE
CC*   UPDATE
CC*   DETLAG
CC*   BLOCK
CC*   WRUB
CC*   WRUDB
CC*
CC* REMARKS
CC*   RESOURCE UTILIZATION BY A CLASS PERFORMING A TASK IS
CC*   DEFINED BY THE RESOURCE UTILIZATION BLOCK (RUB)
CC*   ASSOCIATED WITH THE TASK. THE RUB CONTAINS A LIST OF
CC*   POINTERS TO RESOURCE UTILIZATION DESCRIPTION BLOCKS (RUDES).
CC*   EACH RUDB DEFINES THE RESOURCE INVOLVED, WHETHER RESOURCE
CC*   CONSUMPTION IS DONE BY INDIVIDUALS, BY THE CLASS OR IS A
CC*   FUNCTION OF THE UTILIZATION OF THE PRIMARY RESOURCE.
CC*   THE RUDB ALSO DESCRIBES HOW THE RESOURCE IS CONSUMED AS
CC*   A FUNCTION OF TIME (ARBITRARY OR UNIFORM) AND WHETHER
CC*   SECONDARY AND/OR ALTERNATE RESOURCES EXIST.
CC*   THE CURRENT CODE PERMITS ONE LEVEL OF SECONDARY RESOURCES.
CC*   BOTH PRIMARY AND SECONDARY RESOURCES ARE ALLOWED TO HAVE
CC*   ANY NUMBER OF ALTERNATE RESOURCES.
CC*   WHEN A DEMAND FOR A RESOURCE CANNOT BE SATISFIED AND
CC*   ALTERNATE RESOURCES DO NOT EXIST, THREE USER SELECTED
CC*   OPTIONS CAN BE EXERCIZED: 1.- STOP THE RUN, 2.- CONTINUE
CC*   THE RUN AFTER INDICATING THE RESOURCE SHORTAGE. 3.- LAG
CC*   THE CLASS TO A PERIOD IN TIME WHEN THE MISSING OR SCARCE
CC*   RESOURCE IS AVAILABLE.
CC*
CC* PROGRAMMER
CC*   G. GAIDASZ
CC*   CALSPAN
CC*   AUG 1975
CC*****

```

## RESUSE

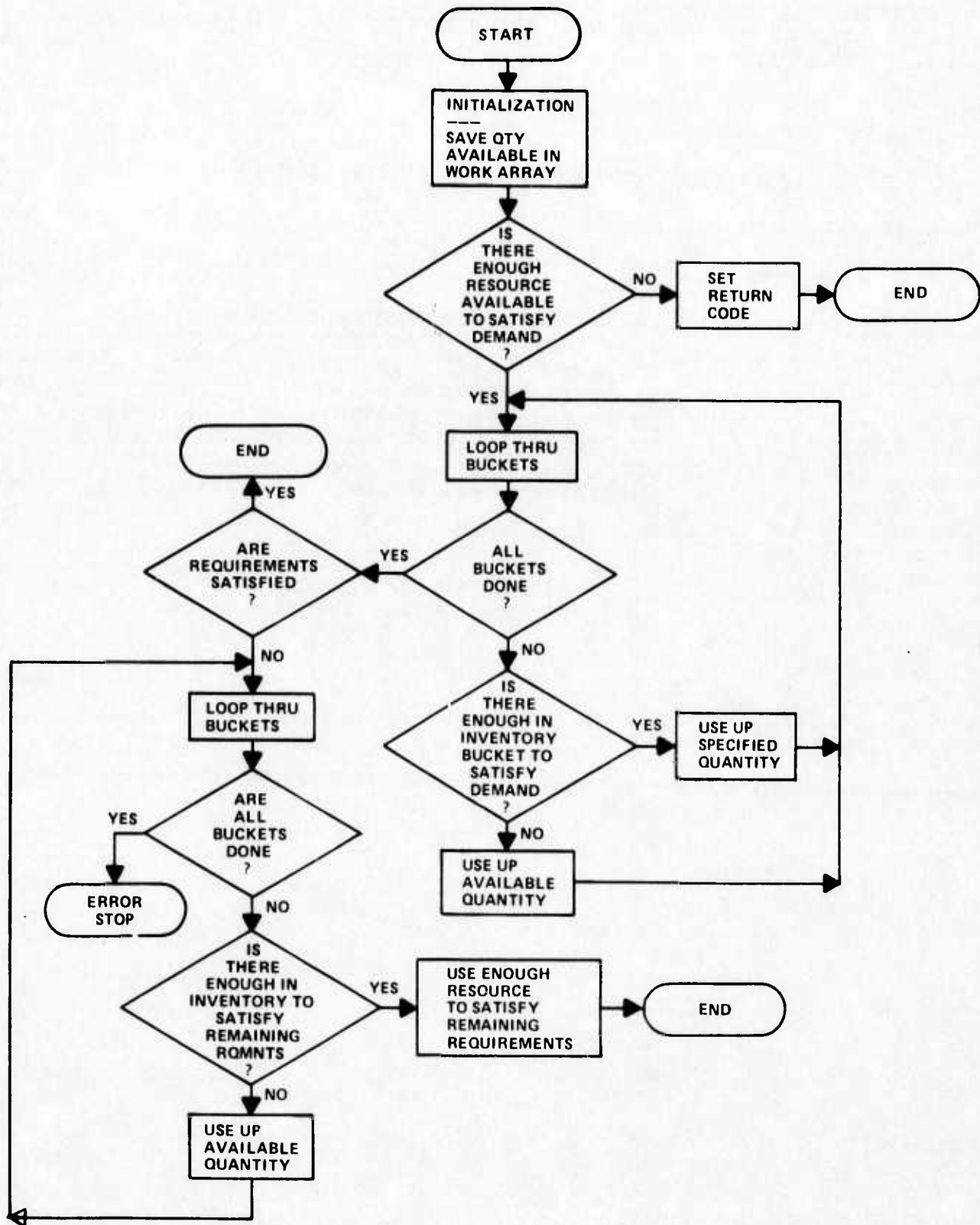


```

C***** RUSER *****
CC*
CC*  PURPOSE
CC*    ALLOCATE RESOURCES FROM INVENTORY TO MEET CALCULATED
CC*    CONSUMPTION.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL RUSER(IQTY,NB,INVRES,I1,I2,ITOTQ, ICODE)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * INPUT *
CC*
CC*    IQTY( )      ARRAY CONTAINING THE RESOURCE QUANTITIES.
CC*                  THAT SHOULD BE CONSUMED IN EACH BUCKET.
CC*    NB          NUMBER OF ENTRIES IN IQTY.
CC*    INVRES( )   RESOURCE INVENTORY ARRAY.
CC*    I1          INDEX OF FIRST PERTINENT ENTRY IN INVRES.
CC*    I2          INDEX OF LAST PERTINENT ENTRY IN INVRES.
CC*    ITOTQ       TOTAL QUANTITY OF RESOURCE REQUIRED TO
CC*                  SATISFY CURRENT DEMAND.
CC*
CC*    * OUTPUT *
CC*
CC*    ICODE        IF 0, ALLOCATION WAS SUCCESSFULL.
CC*                  IF 1, ALLOCATION WAS UNSUCCESSFULL.
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

RUSER



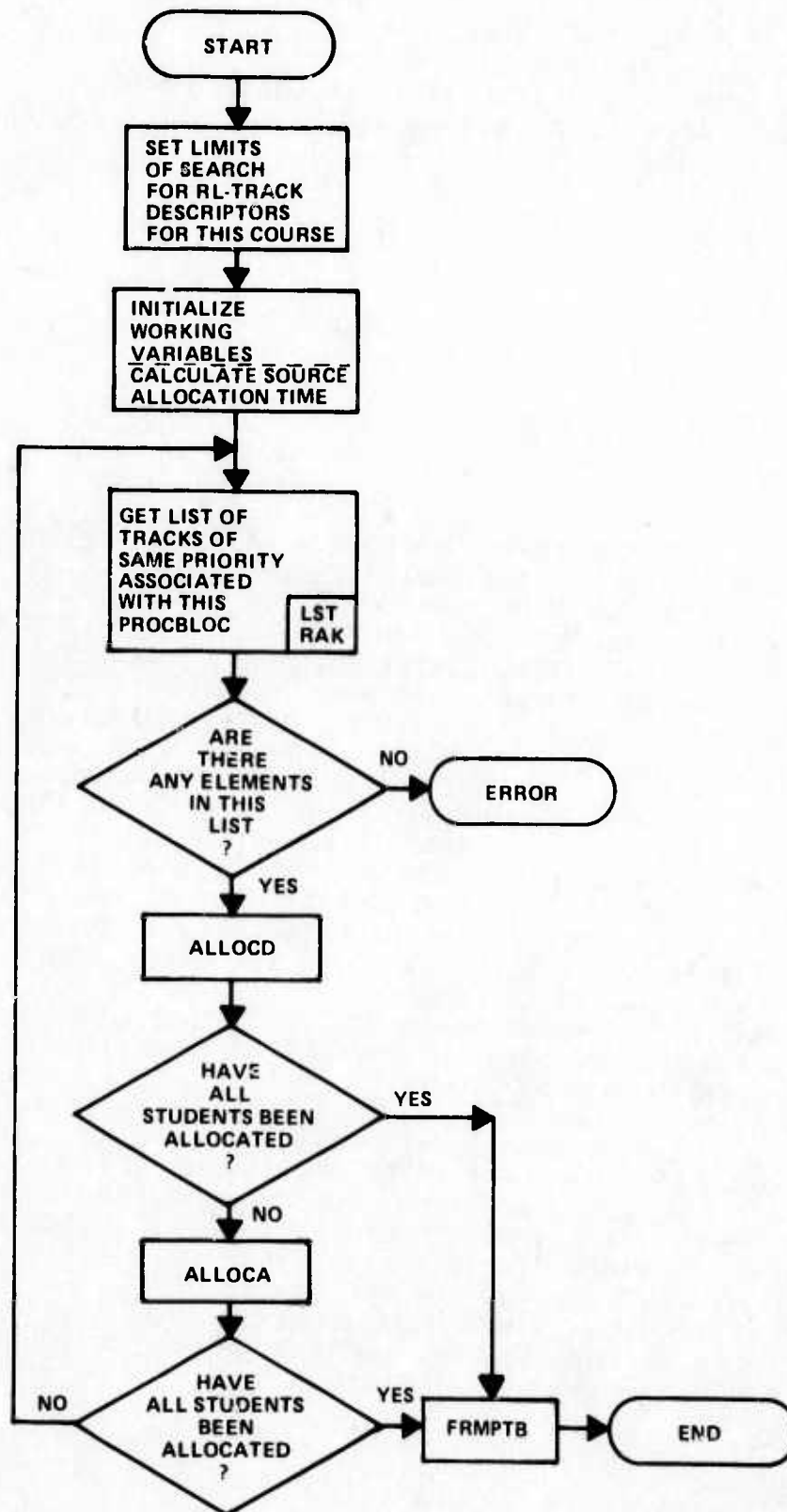
```

CC***** SCATSA *****
CC*
CC*  PURPOSE
CC*    TO SELECT THE TRAINEE SOURCES FOR A CLASS.
CC*    SELECTION IS BASED UPON.
CC*      1. ESTIMATED DURATION OF COURSE.
CC*      2. SOURCE AVAILABILITY AT REQUIRED TIME.
CC*      3. TRANSFER PERCENTAGES AND PRIORITIES SPECIFIED
CC*        FOR THE COURSE.
CC*
CC*  CALLING SEQUENCE
CC*    CALL SCATSA
CC*
CC*  SUBROUTINES USED
CC*    LSTRAK
CC*    ALLOCQ
CC*    ALLOCA
CC*    FRMPTB
CC*    PUTCLS
CC*
CC*  REMARKS
CC*    SCATSA USES THE TRACK DESCRIPTOR BLOCKS CREATED BY TRACKD
CC*    TO LOOK UP SOURCE AVAILABILITY FOR THE TRACKS OF THE COURSE.
CC*    THE TRACK DESCRIPTOR BLOCKS FOR EACH COURSE ARE SORTED
CC*    IN DESCENDING ORDER OF CUMULATIVE PRIORITY BY TRACKD.
CC*    IN SCATSA SUBROUTINE LSTRAK IS USED TO EXTRACT THE POINTERS
CC*    TO A GROUP OF SOURCES THAT HAVE EQUAL PRIORITY.
CC*    SUBROUTINE ALLOCQ IS THEN USED TO TRY TO ALLOCATE THE
CC*    PROPER NUMBER OF TRAINEES TO EACH TRACK AS SPECIFIED
CC*    BY THE TRANSFER PROPORTIONS SPECIFIED FOR THE COURSE.
CC*    IF ALL TRAINEES IN THE CLASS HAVE BEEN ALLOCATED BY
CC*    TRACKD, SUBROUTINE FRMPTB IS INVOKED.
CC*    IF THE DESIRED PROPORTIONING COULD NOT BE SATISFIED BY
CC*    ALLOCQ THEN SUBROUTINE ALLOCA IS CALLED. ALLOCA LOOPS
CC*    ON THE SAME PRIORITY TRACKS AS ALLOCQ, BUT IT ALLOCATES
CC*    AS MANY STUDENTS AS POSSIBLE TO THE TRACKS THAT HAVE ANY
CC*    REMAINING SOURCE INVENTORY.
CC*    IF THE CLASS CANNOT BE ALLOCATED TO THE SOURCES OF THIS
CC*    PRIORITY GROUP, THEN THE PROCESS IS REPEATED USING THE
CC*    SET OF NEXT LOWER PRIORITY TRACKS.
CC*    AFTER ALL TRAINEES HAVE BEEN ALLOCATED TO THEIR RESPECTIVE
CC*    SOURCES, SUBROUTINE FRMPTB IS INVOKED.
CC*    THE PURPOSE OF FRMPTB IS TO CREATE THE PREDETERMINED
CC*    TRANSFER BLOCKS (PTBS) FOR THE CLASS BEING PROCESSED.
CC*    THE PTBS FORM A TREE STRUCTURE THAT ORIGINATES AT THE
CC*    PROCBLOC FROM WHICH SCATSA WAS INVOKED AND HAS A LINKED
CC*    ELEMENT FOR EACH NODE OF THE TRACKS TO THE LEFT OF IT.
CC*    THE PTB INDICATES HOW MANY STUDENTS SHOULD TAKE EACH ONE
CC*    OF THE FIVE POSSIBLE BRANCHES AT EACH NODE AND A POINTER
CC*    TO THE NEXT NODE (IF ANY) ALONG EACH ONE OF THE FIVE
CC*    BRANCHES.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*****

```



# SCATSA



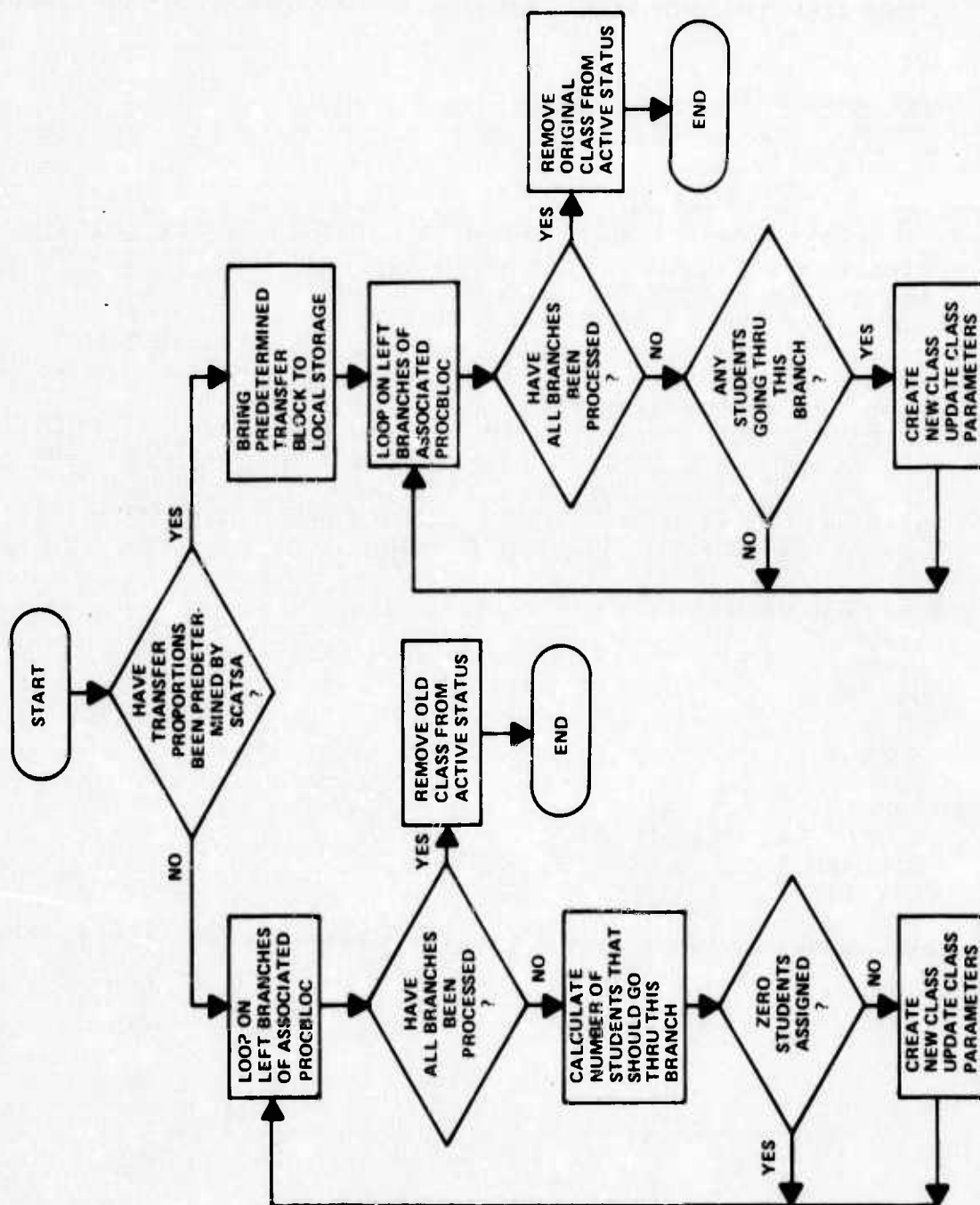


```

CC***** SPLIT *****
CC*
CC*  PURPOSE
CC*    TO SPLIT A CLASS INTO MULTIPLE CLASSES.
CC*    IF SUB. SCATSA HAS BEEN ALREADY EXECUTED FOR THIS CLASS
CC*    THEN THE PREDETERMINED TRANSFER PROPORTIONS WILL BE USED,
CC*    OTHERWISE THE PROCBLOC TRANSFER PROPORTIONS WILL BE USED.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SPLIT
CC*
CC*  REMARKS
CC*    IF SCATSA HAS NOT BEEN EXECUTED FOR THIS TRACK, SPLIT
CC*    CALCULATES THE NUMBER OF STUDENTS THAT SHOULD BE SENT
CC*    ALONG EACH BRANCH FROM THE PROPORTIONS SPECIFIED FOR
CC*    EACH LEFT BRANCH OF THE ACTIVE PROCBLOC.
CC*    IF SCATSA HAS BEEN EXECUTED FOR THIS TRACK THEN THE
CC*    NUMBER OF STUDENTS TO BE SENT ALONG EACH BRANCH IS TAKEN
CC*    FROM THE APPROPRIATE PTB.
CC*    AFTER THE CLASS HAS BEEN SPLIT INTO THE DESIRED PROPORTIONS,
CC*    NEW CLASSES ARE CREATED FROM EACH OF THE NEW GROUPS AND
CC*    THE OLD CLASS IS RELEASED. THE UNIQUE CLASS NUMBER IS
CC*    ASSIGNED TO EACH OF THE NEW CLASSES AND THE CORRECT PTB
CC*    ADDRESS IS ENTERED. IF APPROPRIATE
CC*
CC*  SUBROUTINES USED
CC*    CBLOCK
CC*    REMCLS
CC*    GETPTB
CC*    WPTB
CC*    NEWCLS
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# SPLIT

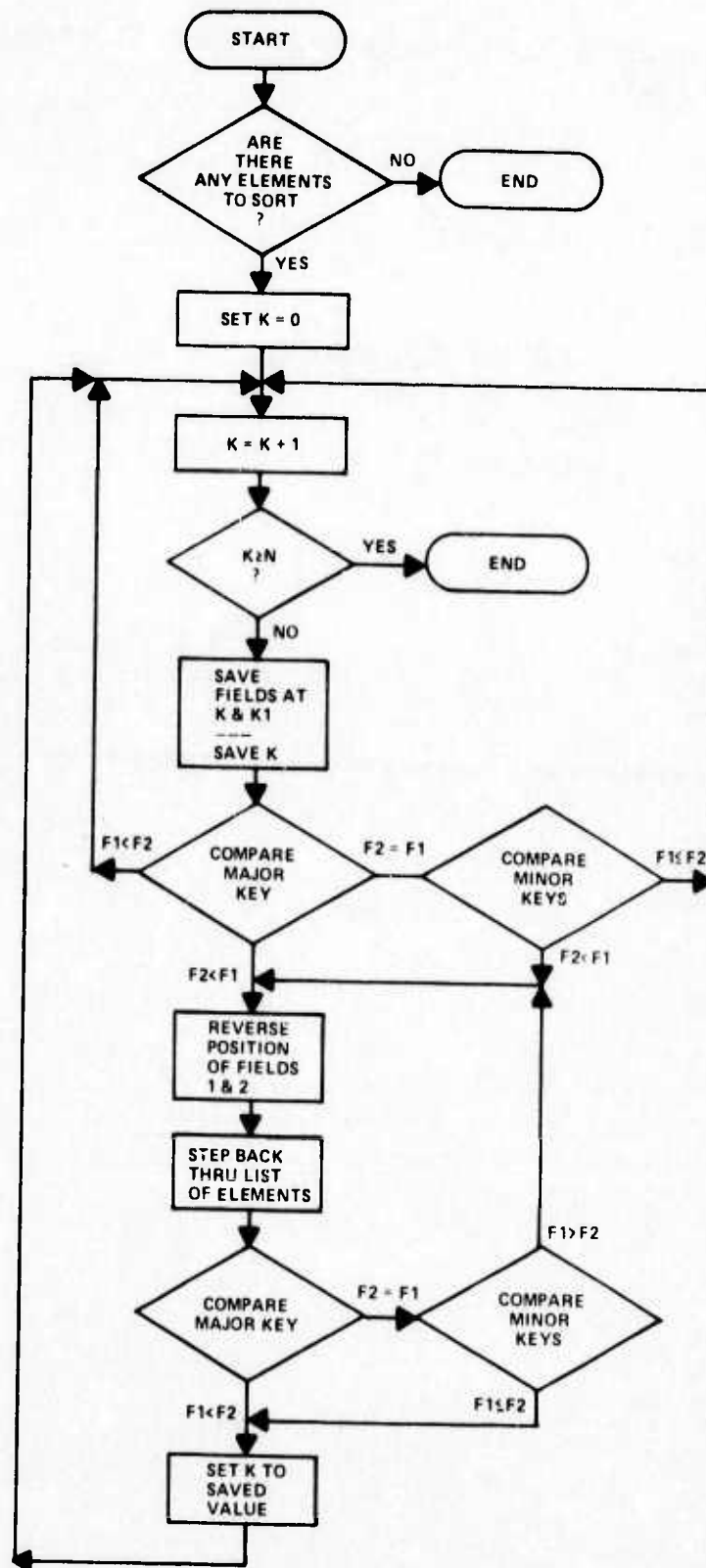


```

CC***** SRTCTP *****
CC*
CC*  PURPOSE
CC*    UTILITY ROUTINE TO SORT CLASSES IN ORDER BY GRADUATION
CC*    DATE AND PRIORITY.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SRTCTP(IT1,IT2,N,IA)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IT1()      ARRAY OF MAJOR KEYS.
CC*    IT2()      ARRAY OF MINOR KEYS.
CC*    N          NUMBER OF ELEMENTS TO BE SORTED.
CC*    IA()       ARRAY OF POSITION POINTERS TO SORTED RECORDS.
CC*
CC*  REMARKS
CC*    METHOD IS AN INDEX BUBBLE SORT.
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# SRTCTP

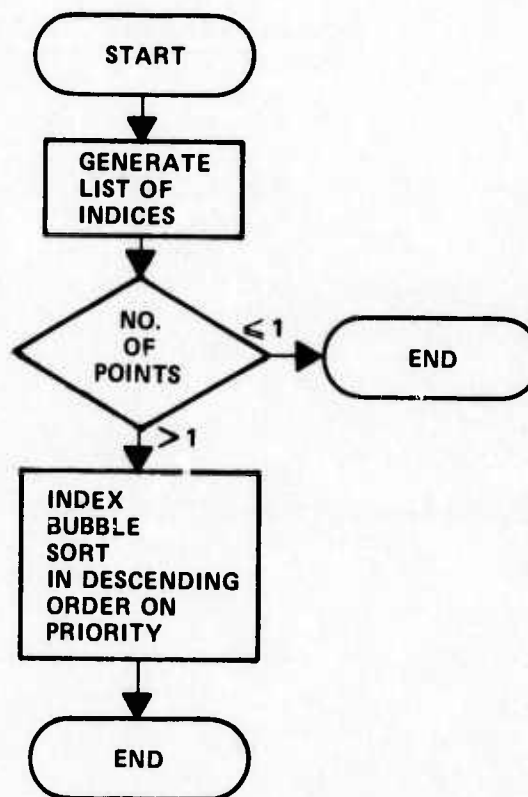


```

CC***** SRTTDB *****
CC*
CC* PURPOSE
CC*   TO DO AN INDEX SORT ON THE TRACK DESCRIPTOR BLOCKS FOR
CC*   A COURSE. THE SORT IS IN DESCENDING ORDER ON PRIORITY.
CC*
CC* CALLING SEQUENCE
CC*
CC*   CALL SRTTDB(I1,I2,CUMPTY,ITDBST)
CC*
CC* DESCRIPTION OF PARAMETERS
CC*
CC*   * INPUT *
CC*
CC*   I1           POINTER TO FIRST ELEMENT TO BE SORTED.
CC*   I2           POINTER TO LAST ELEMENT TO BE SORTED.
CC*   CUMPTY()     SORT FIELD.
CC*   * I/O *
CC*
CC*   ITDBST()     INDEX OF SORTED ELEMENTS
CC*
CC* REMARKS
CC*   METHOD IS A INDEX BUBBLE SORT.
CC*
CC* PROGRAMMER
CC*   G. GAIDASZ
CC*   CALSPAN
CC*   AUG 1975
CC*
CC*****

```

# SRTTDB



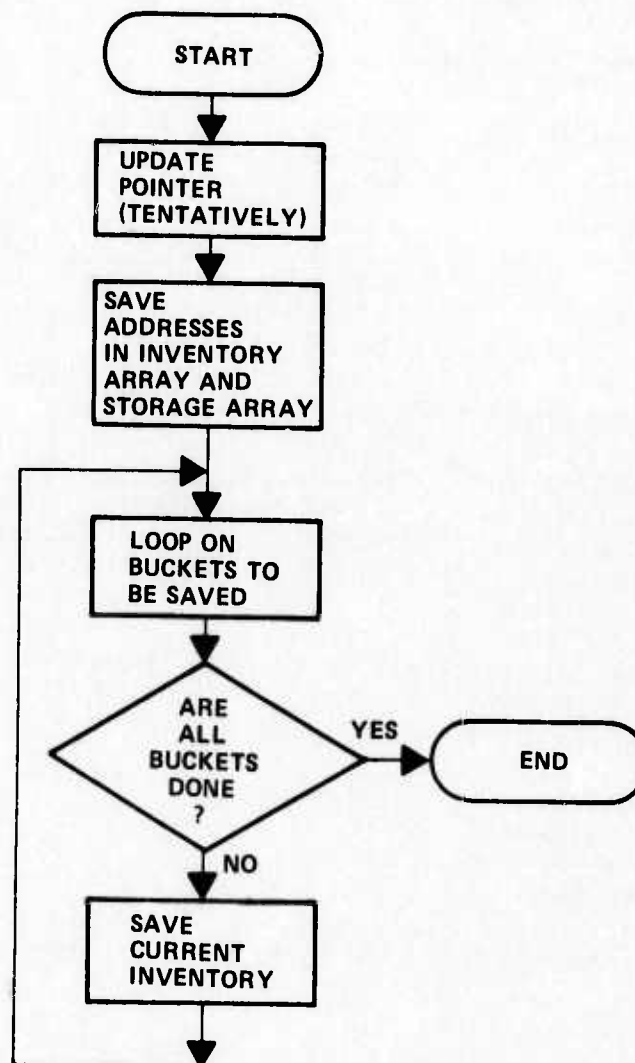


```

CC***** SVRUS1 *****
CC*
CC*  PURPOSE
CC*    SAVE PERTINENT INVENTORY OF THE RESOURCE BEING PROCESSED.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SVRUS1
CC*
CC*  DESCRIPTION OF PARAMETERS.
CC*
CC*    * IMPLICIT INPUT VIA COMMON *
CC*
CC*    INDX1          POINTER TO ELEMENT IN ARRAY INVRES THAT
CC*                  CORRESPONDS TO THE FIRST BUCKET THAT MAY
CC*
CC*    INDX2          POINTER TO ELEMENT IN ARRAY INVRES THAT
CC*                  CORRESPONDS TO THE LAST BUCKET THAT MAY
CC*                  BE USED BY THE CURRENT CLASS.
CC*
CC*    NSAVE          NUMBER OF RESOURCE INVENTORIES SAVED UP
CC*                  TO NOW.
CC*
CC*    ISAVE          TOTAL NUMBER OF BUCKETS SAVED UP TO NOW.
CC*
CC*    * IMPLICIT OUTPUT VIA COMMON *
CC*
CC*    IADI1()        SAVED VALUE OF INDX1
CC*    IADI2()        SAVED VALUE OF INDX2
CC*    IADS1()        POINTER TO FIRST ELEMENT SAVED IN IAUSED.
CC*    IAUSED()       SAVED RESOURCE INVENTORIES.
CC*
CC*  REMARKS
CC*    THIS ROUTINE SAVES THE PERTINENT RESOURCE INVENTORY BEFORE
CC*    THE CURRENT UPDATES ARE MADE. SVRUS2 IS LATER CALLED IF
CC*    NECESSARY TO CALCULATE THE ACTUAL CONSUMPTION.
CC*
CC*  PROGRAMMER
CC*    G.GAIDASZ
CC*    AUG 1975
CC*    CALSPAN
CC*
CC*****

```

SVRUS1

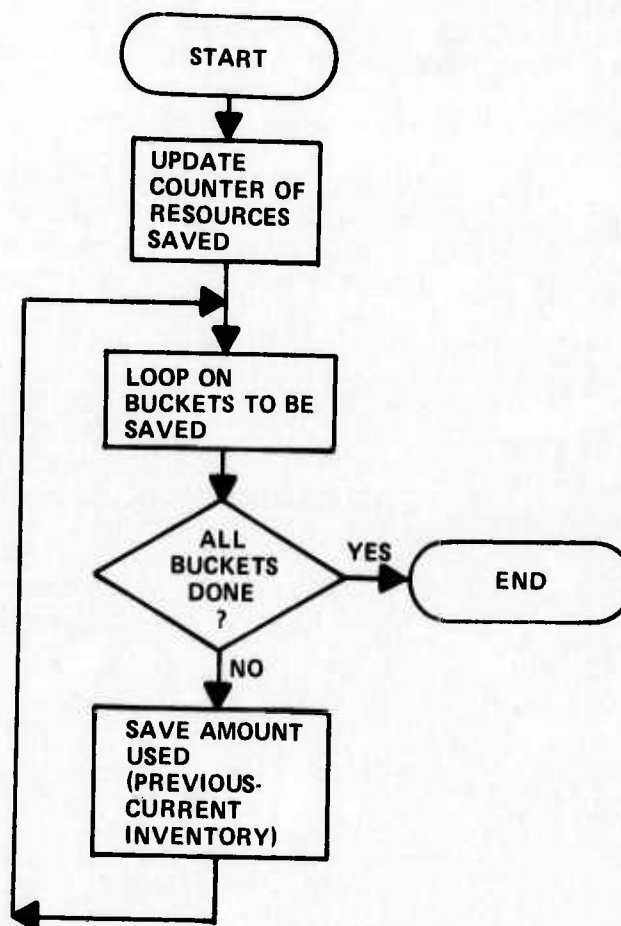


```

CC***** SVRUS2 *****
CC*
CC*  PURPOSE
CC*    CALCULATE ACTUAL USAGE OF CURRENT RESOURCE BY SUBTRACTING
CC*    CURRENT INVENTORY FROM PREVIOUS INVENTORY.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SVRUS2
CC*
CC*  DESCRIPTION OF PARAMETERS.
CC*    SAME AS IN SVRUS1, EXCEPT THAT IAUSED() IS UPDATED TO
CC*    INDICATE CONSUMPTION BY SUBTRACTING THE CURRENT INVENTORY
CC*    FROM IT
CC*
CC*  REMARKS
CC*    SVRUS2 IS ONLY INVOKED FOR SATISFIED PRIMARY COMPOSITE
CC*    RESOURCES AND THEIR ASSOCIATED SECONDARIES. RESTORATION
CC*    OF OTHER RESOURCES (IN CASE OF RESOURCE ALLOCATION
CC*    FAILURE) IS HANDLED BY NOT CALLING UPDATE.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

SVRUS2

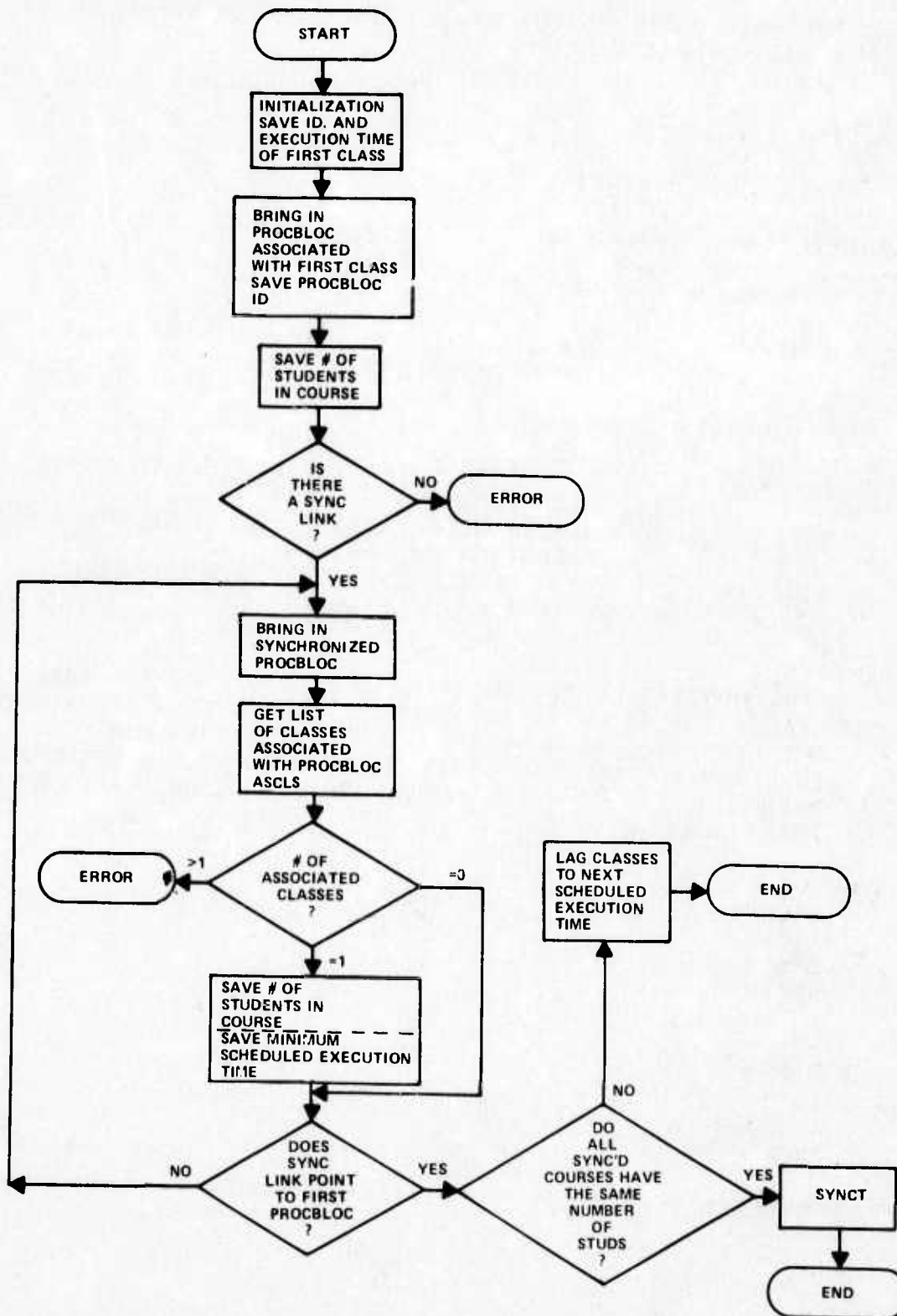


```

CC***** SYNC *****
CC*
CC*  PURPOSE
CC*    TO FORCE SIMULTANEOUS PROCESSING (SYNCHRONIZATION) OF
CC*    A LIST OF PROCBLOCKS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SYNC(MINTME)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    * OUTPUT *
CC*
CC*    MINTME          TIME TO WHICH CLASSES WILL BE LAGGED IF
CC*                   SYNCHRONIZATION IS IMPOSSIBLE AT THIS TIME.
CC*
CC*    * IMPLICIT OUTPUT *
CC*
CC*    IFAIL          IS SET TO 1 IF SYNCHRONIZATION CANNOT BE
CC*                   ACCOMPLISHED.
CC*
CC*    NOCLS          NUMBER OF CLASSES TO BE SYNCHRONIZED IN
CC*                   EXECUTION OR LAGGED.
CC*
CC*    INDXC()        LIST OF CLASSES TO BE SYNCHRONIZED IN
CC*                   EXECUTION OR LAGGED.
CC*
CC*  REMARKS
CC*    SYNCHRONIZATION REQUIRES THAT CLASSES WITH THE SAME
CC*    GRADUATION ID BE IN THE CORRECT PROCBLOC(S) OF EACH OF
CC*    THE COURSES LINKED BY THE SYNCHRONIZATION LOOP.
CC*    IF CLASSES HAVE BEEN SPLIT THEN THE NUMBER OF STUDENTS
CC*    (WITH THE SAME GRADUATION ID) IN EACH COURSE (SUMMED OVER
CC*    THE TRACKS) MUST AGREE.
CC*
CC*  SUBROUTINES USED
CC*    PBLOCK
CC*    ASCLS
CC*    SYNCT
CC*    BLOCK
CC*    CLSDMP
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# SYNC

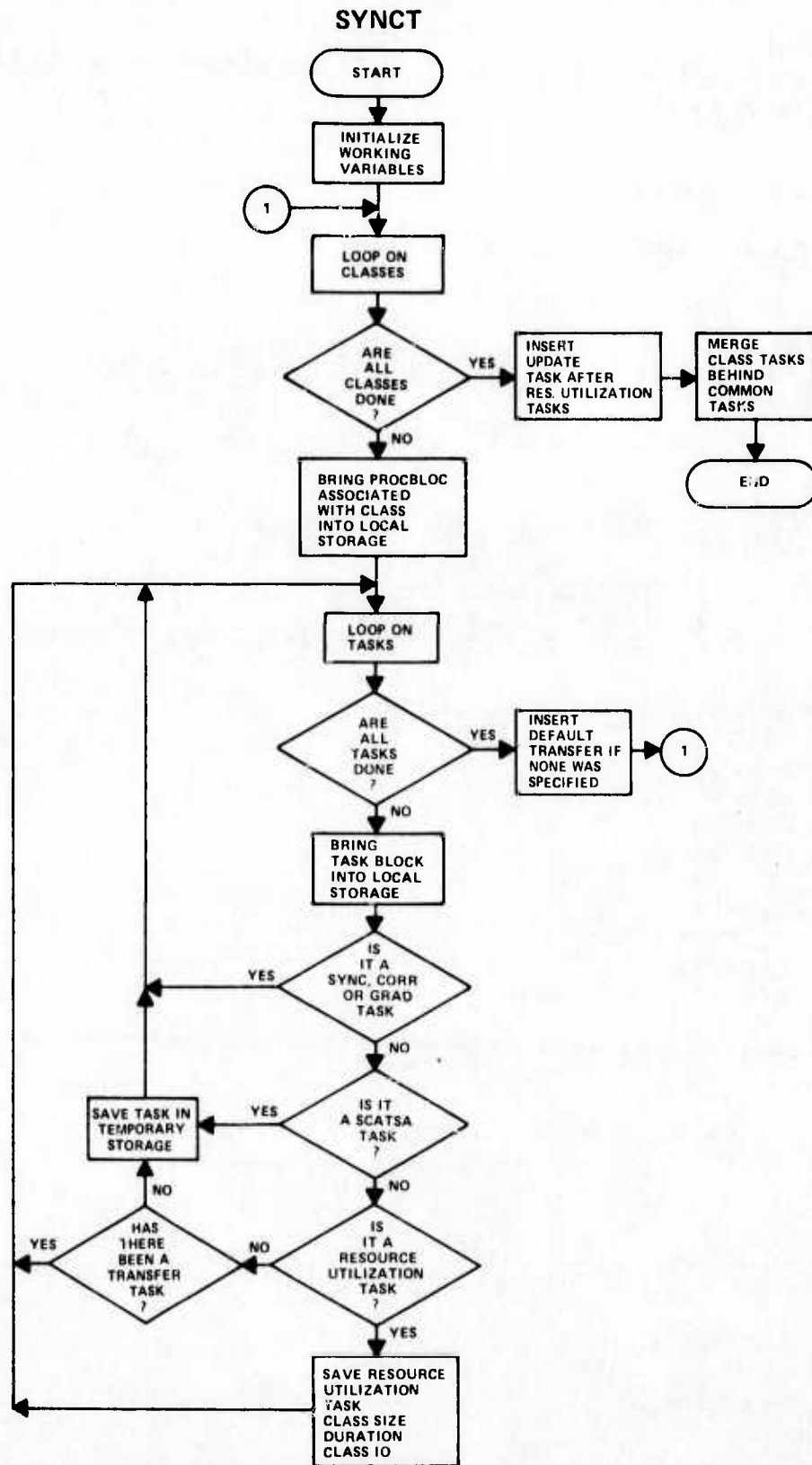




```

CC***** SYNCT *****
CC*
CC*  PURPOSE
CC*    TO PLACE ALL TASKS THAT MUST BE PERFORMED SIMULTANEOUSLY
CC*    IN A LIST.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL SYNCT(NOCLS,INDXC)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    NOCLS          NUMBER OF CLASSES IN INDXC.
CC*    INDXC()        LIST OF POINTERS TO CLASSES IN THE
CC*                   CURRENT CLASSES ARRAYS, WHOSE TASKS ARE
CC*                   TO BE SYNCHRONIZED.
CC*
CC*  REMARKS
CC*    THE TASKS ARE ARRANGED IN THE FOLLOWING ORDER:
CC*      1. RESOURCE UTILIZATION TASKS.
CC*      2. UPDATE TASK. (PROVIDED BY PROGRAM)
CC*      3. OTHER TASKS.
CC*      4. TRANSFER TASKS (PROVIDED BY PROGRAM).
CC*
CC*  SUBROUTINES USED
CC*    PLIST
CC*    PBLOCK
CC*    BLOCK
CC*    GETCLS
CC*    TBLOCK
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

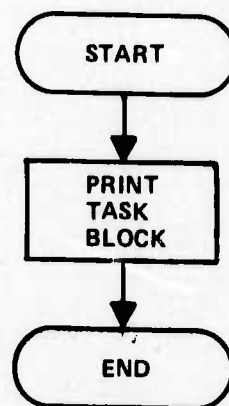


```

CC***** TBLOCK *****
CC*
CC*  PURPOSE
CC*    TO PRINT A TASK BLOCK
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL TBLOCK(IADRS,IBLOCK)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS          ADRESS OF TASK BLOCK.
CC*    IBLOCK        FIRST WORD OF TASK BLOCK.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# TBLOCK

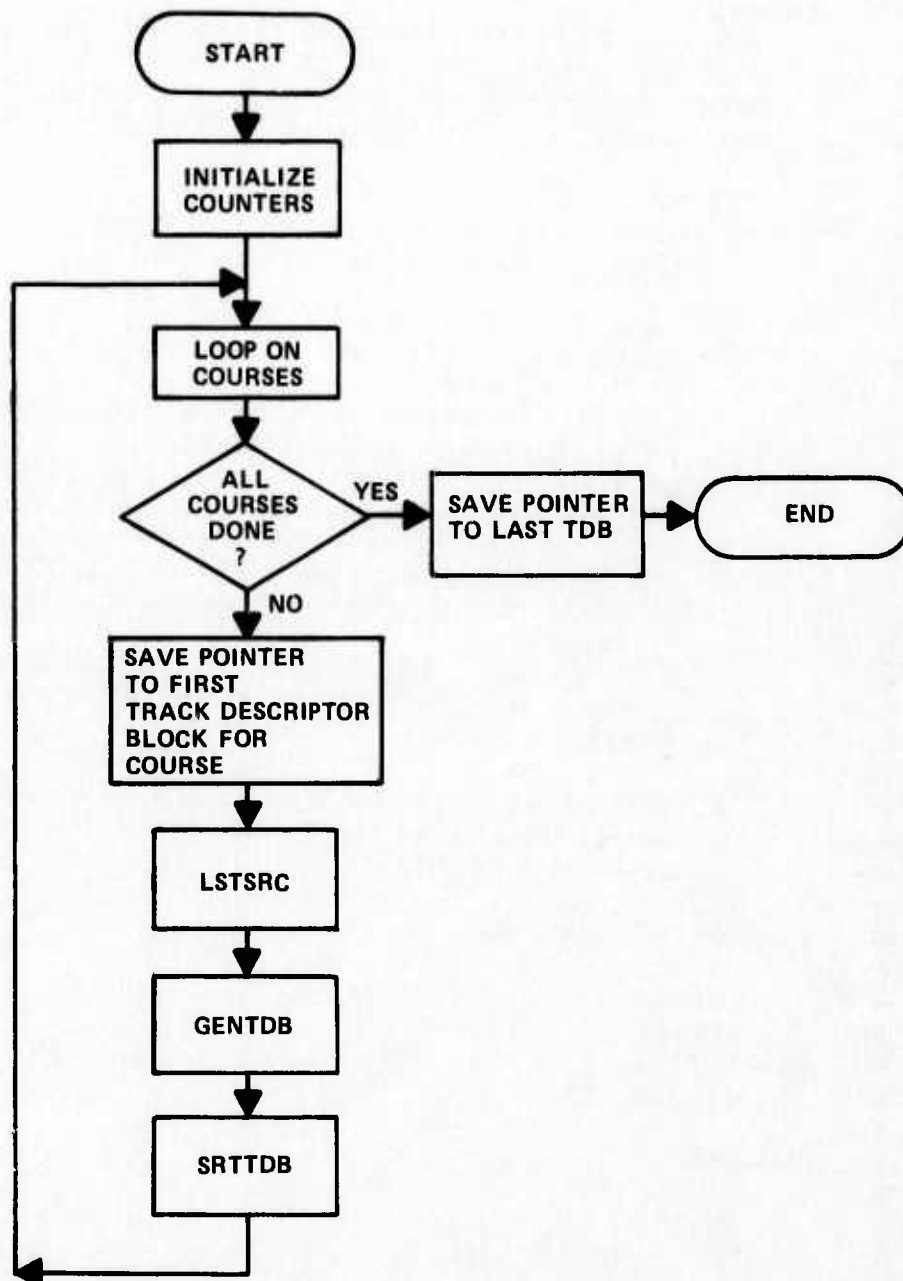


```

CC***** TRACKD *****
CC*
CC*  PURPOSE
CC*    TO CREATE THE TRACK DESCRIPTOR BLOCKS FOR EACH COURSE.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL TRACKD
CC*
CC*  REMARKS
CC*    THE CREATION OF THE TRACK DESCRIPTOR BLOCKS, WHICH ARE
CC*    USED BY SUBROUTINE SCATSA TO CHOOSE TRAINEE SOURCES
CC*    ACCORDING TO THE PRIORITY AND PROPORTIONS SPECIFIED IN
CC*    THE COURSE DESCRIPTION, IS ACCOMPLISHED IN THREE STEPS.
CC*
CC*      1. A RIGHT TO LEFT SCAN IS DONE OF THE PROCBLOCKS IN
CC*          EACH COURSE. THE RESULT OF THIS OPERATION IS A
CC*          LIST OF SOURCES FOR EACH COURSE.
CC*
CC*      2. EACH SOURCE PRODUCED BY STEP 1 SERVES AS A STARTING
CC*          POINT FOR A LEFT TO RIGHT SCAN OF THE TRACK DEFINED
CC*          BY THE SOURCE.
CC*          AS THE PROCBLOCKS OF A TRACK ARE PROCESSED A COUNT
CC*          IS KEPT OF THE TOTAL DURATION OF THE TRACK.
CC*          PROPORTIONS SPECIFIED ALONG THE TRACK ARE MULTIPLIED
CC*          TOGETHER TO PRODUCE THE RESULTING PROPORTION AT EACH
CC*          NODE (PROCBLOCK WITH MORE THAN ONE LEFT BRANCH). THE
CC*          PRIORITY OF EACH TRACK IS CALCULATED BY ADDING THE
CC*          CURRENT PRIORITY TO THE PREVIOUS PRIORITY DIVIDED BY
CC*          A HUNDRED.
CC*          A TRACK DESCRIPTOR BLOCK IS CREATED FOR EACH NODE
CC*          ALONG A TRACK. EACH NODE POINTS TO THE NODE THAT
CC*          PRECEDES IT.
CC*
CC*      3. THE FINAL STEP (SRTTDB) IS A LINK SORT IN DECREASING
CC*          ORDER BY PRIORITY OF THE TRACK DESCRIPTOR BLOCKS
CC*          FOR EACH COURSE.
CC*
CC*  SUBROUTINES CALLED
CC*
CC*    LSTSRC
CC*    GENTDB
CC*    SRTTDB
CC*    WRLTDB
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

TRACKD



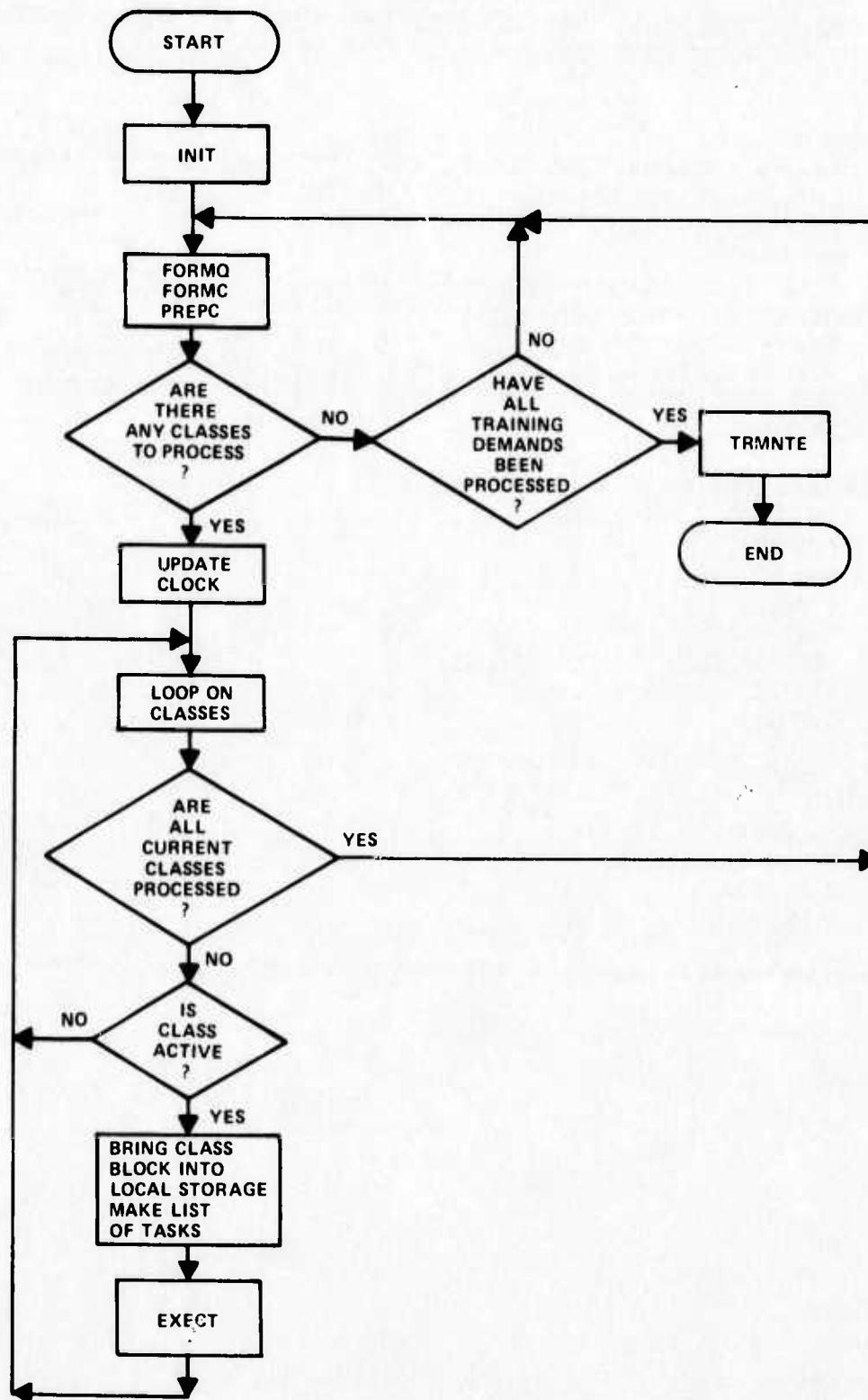


```

CC***** TRAM3 *****
CC*
CC*  PURPOSE
CC*    PROVIDE OVER-ALL CONTROL LOGIC FOR THE CALCULATION OF
CC*    RESOURCES USED BY 'CLASSES' OF STUDENTS GOING THROUGH
CC*    USER DEFINED COURSES.
CC*
CC*  REMARKS
CC*    TRAM3 PROVIDES THE BASIC CYCLING LOOP OF THE PROGRAM.
CC*    AFTER INITIALIZATION IS DONE, THE PROGRAM READS TRAINING
CC*    DEMANDS TO CALCULATE THE NUMBER OF TRAINEES THAT SHOULD
CC*    BE GRADUATED.
CC*    FROM THE TRAINING DEMANDS CLASSES ARE STORED IN MASS
CC*    STORAGE. SUBROUTINE PREPC IS USED TO SELECT THE CLASSES
CC*    THAT SHOULD BE ACTIVE FOR THE CURRENT SIMULATION TIME.
CC*    THE PROGRAM THEN LOOPS OVER THESE ACTIVE CLASSES, SELECTING
CC*    AND EXECUTING THE TASKS SPECIFIED BY THE PERTINENT
CC*    PROCBLOCS.
CC*
CC*  SUBROUTINES USED
CC*    INIT
CC*    FORMQ
CC*    PREPC
CC*    TRMNT
CC*    CLOCK
CC*    CBLOCK
CC*    EXECT
CC*    GETCLS
CC*    LSTASK
CC*    FORMC
CC*
CC*  PROGRAMMER
CC*    GEORGE GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

# TRAM3

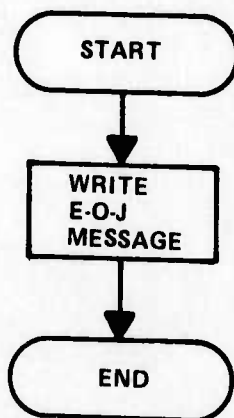


```

CC***** TRMNT *****
CC*
CC*  PURPOSE
CC*    INDICATE TERMINATION OF TRAM-3 EXECUTION.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL TRMNT
CC*
CC*  PROGRAMMER
CC*    G.GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

TRMNT

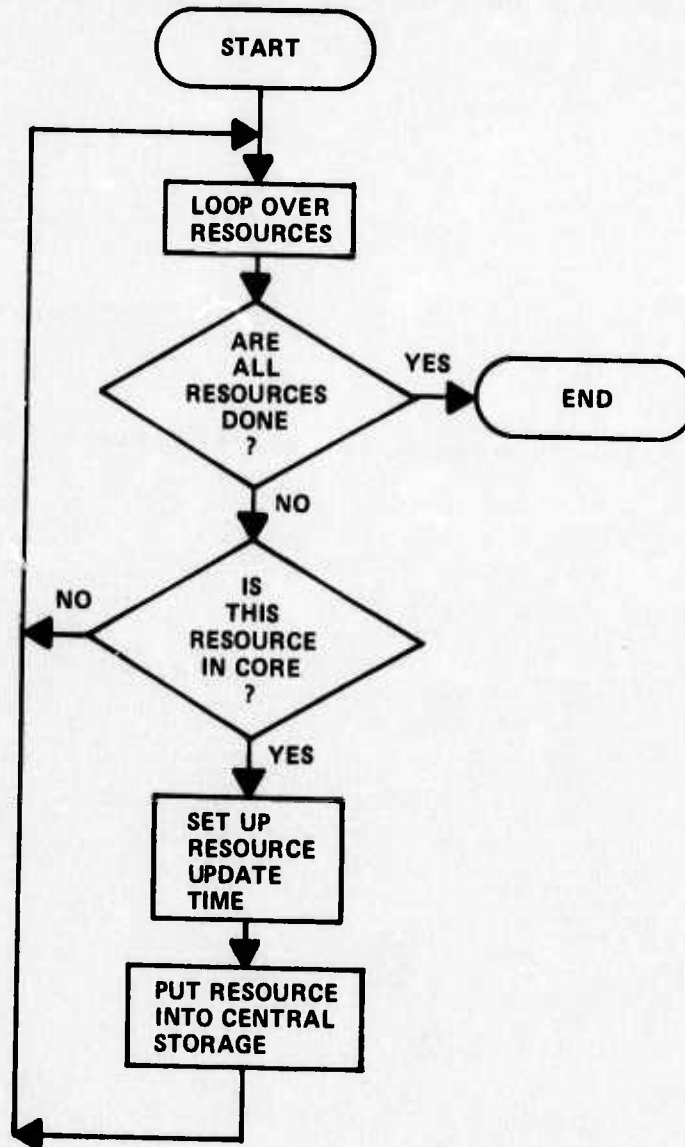


```

CC***** UPDATE *****
CC*
CC*  PURPOSE
CC*    UPDATE RESOURCE INVENTORIES TO REFLECT THE CONSUMPTION
CC*    BY THE CURRENT TASK OR GROUP OF SYNCHRONIZED TASKS.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL UPDATE
CC*
CC*  SUBROUTINES USED
CC*    PUTRES
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    AUG 1975
CC*
CC*****

```

# UPDATE

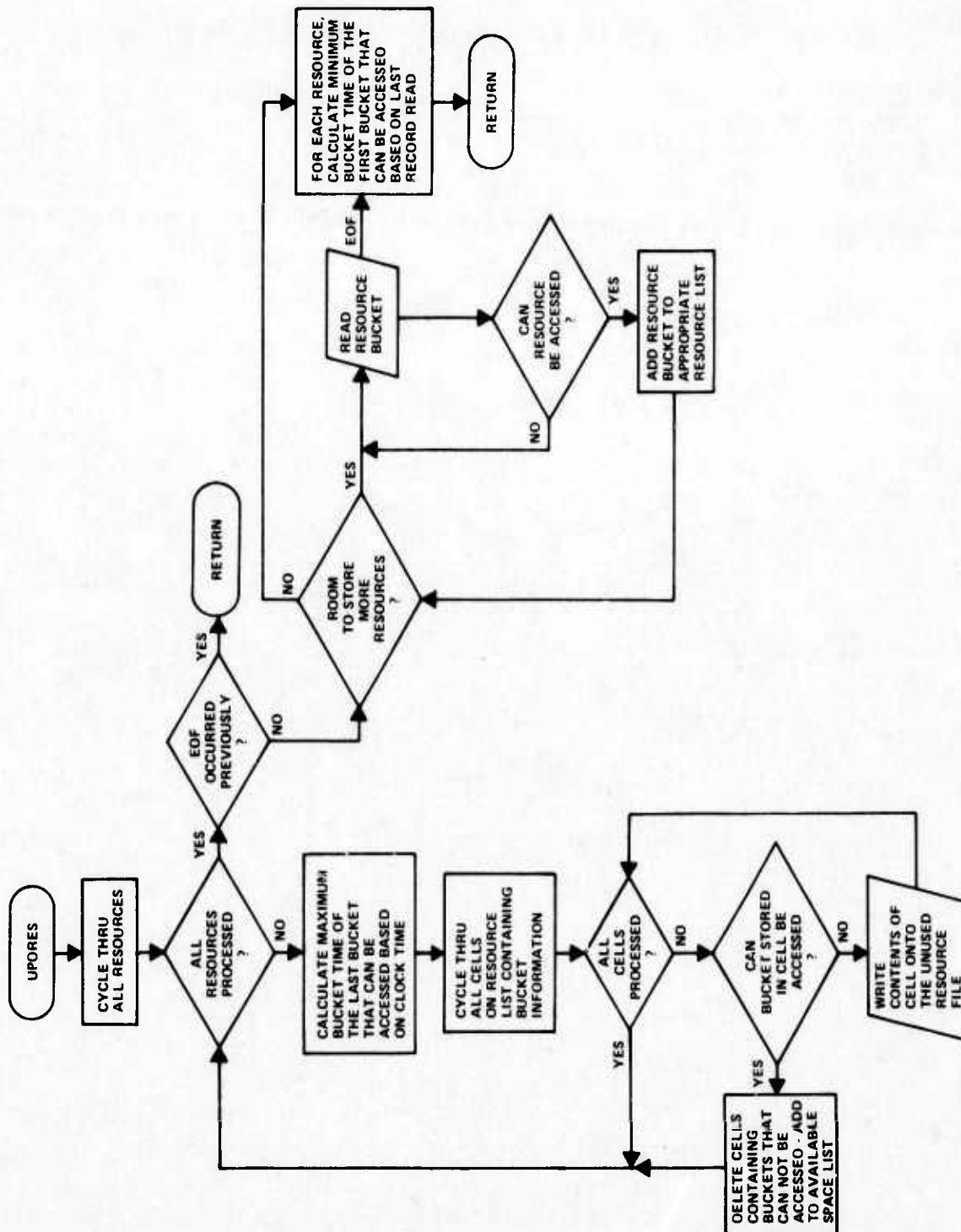




```

C***** UPDRES *****
C*
C*          SUBROUTINE UPDRES
C*
C*  PURPOSE
C*    UPDATES RESOURCE LISTS WHEN CLOCK TIME HAS CHANGED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*
C*****

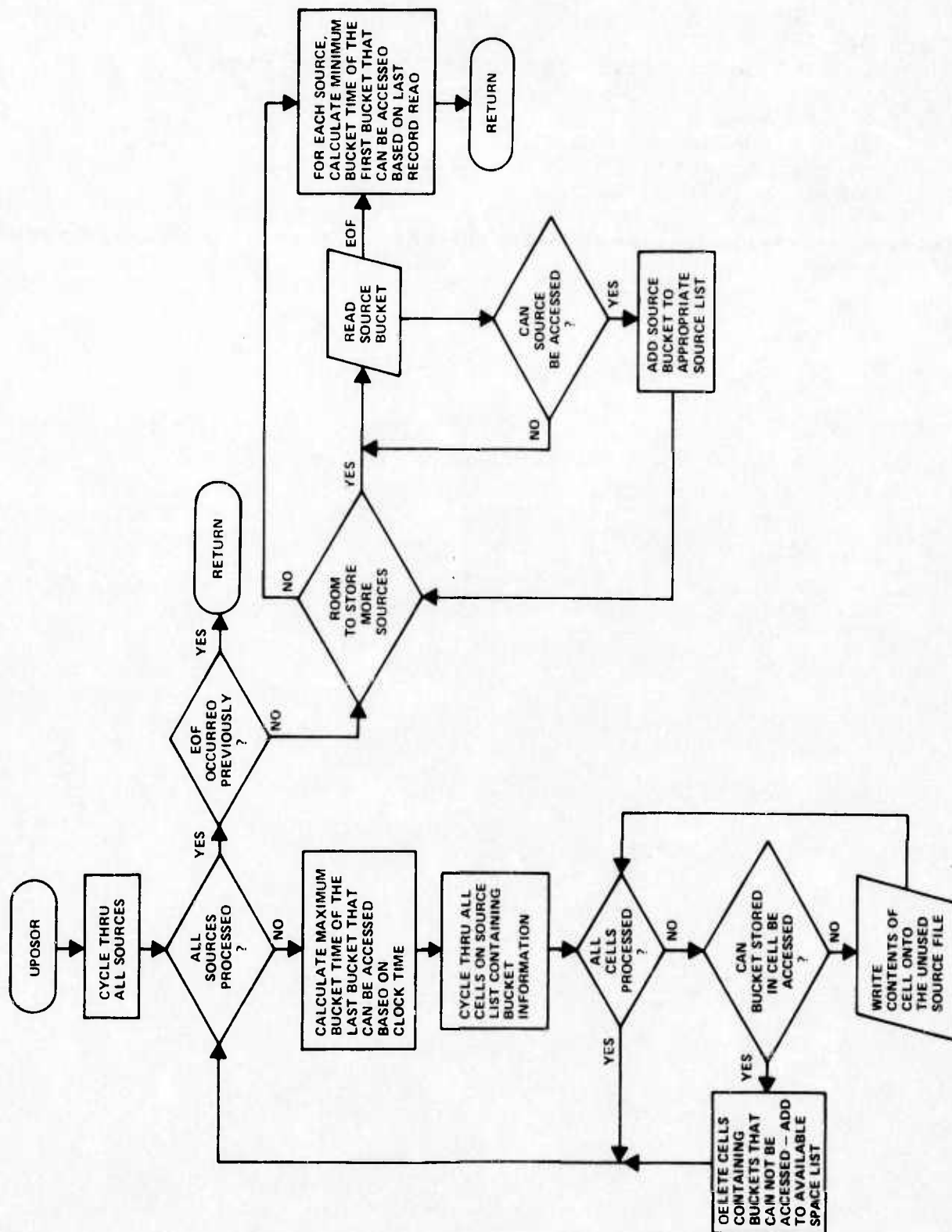
```



```

C***** UPDSOR *****
C*
C*          SUBROUTINE UPDSOR
C*
C*  PURPOSE
C*    UPDATES SURCE LISTS WHEN CLOCK TIME HAS CHANGED
C*
C*  AUTHOR/PROGRAMMER
C*    JOHN R. MENIG
C*    CALSPAN CORPORATION
C*    28 APRIL 1975
C*
C*****

```

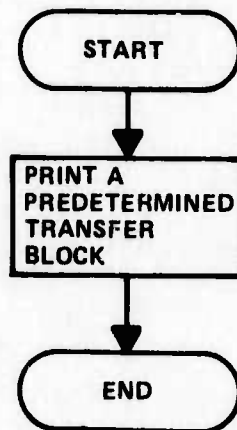


```

CC***** WPTB *****
CC*
CC* PURPOSE
CC*   TO PRINT A PREDETERMINED TRANSFER BLOCK.
CC*
CC* CALLING SEQUENCE
CC*
CC*   CALL WPTB(IC,IP,N,NXT)
CC*
CC* DESCRIPTION OF PARAMETERS
CC*
CC*       IC           ADRESS OF CLASS ASSOCIATED WITH PTB.
CC*       IP           ADRESS OF PROCBLOC ASSOCIATED WITH PTB.
CC*       N()          NUMBER OF STUDENTS TO BE SENT ALONG EACH
CC*                   OF THE 5 BRANCHES OF THE PTB.
CC*       NXT()        ADDRESSES OF THE NEXT PTBS ALONG EACH OF
CC*                   THE 5 BRANCHES.
CC*
CC* PROGRAMMER
CC*   G. GAIDASZ
CC*   CALSPAN
CC*   MAY 1975
CC*
CC*****

```

WPTB



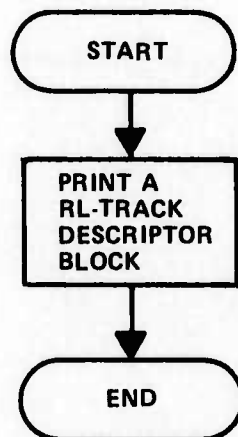


```

CC***** WRLTDB *****
CC*
CC*  PURPOSE
CC*    TO PRINT A RL-TRACK DESCRIPTOR BLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL WRLTDB(I)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    I                INDEX OF TRACK DESCRIPTOR BLOCK TO BE PRINTED
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

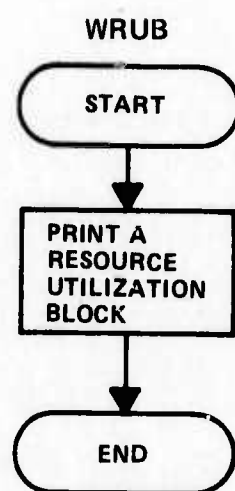
**WRLTDB**



```

CC***** WRUB *****
CC*
CC*  PURPOSE
CC*    TO PRINT A RESOURCE UTILIZATION BLOCK
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL WRUB(IADRS,IBLOCK)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS          ADRESS OF RUB.
CC*    IBLOCK()      FIRST WORD OF RUB.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

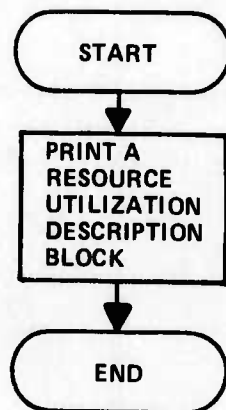


```

CC***** WRUDB *****
CC*
CC*  PURPOSE
CC*    TO PRINT A RESOURCE UTILIZATION DESCRIPTION BLOCK.
CC*
CC*  CALLING SEQUENCE
CC*
CC*    CALL WRUDB(IADRS,IBLOCK)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*
CC*    IADRS      ADRESS OF RUDB.
CC*    IBLOCK     FIKST WORD OF RUDB.
CC*
CC*  PROGRAMMER
CC*    G. GAIDASZ
CC*    CALSPAN
CC*    MAY 1975
CC*
CC*****

```

WRUDB





### 3.5 Cross Reference Table

In the table on the following pages, the column headings show the subroutine names that do the calling, and the row heading give the names of the subroutines called.

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAMB \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	MAIN	SRCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	IBLOCK	TRACKD	TRMTE	UPDATE	WPTB
ADDTG												
ALLOC												
ALLOCA												
ALLOCD												
ASCLS						X						
ASCLSS												
CALQ												
CBLOCK	X											
CLASCG												
CLSDMP						X						
CORR												
DETLG												
DTRNSF												
ERROR	X			X		X		X				
EXECT	X											
FORMC	X											
FORMQ	X											
FRETDB												
FRMPTB												
GENTDB									X			
GETCLS	X						X					
GLTPTB												
GETTOB												
GRADE												
INIT	X											
INITR												
LAG												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	WRLTDB	WRUB	WRUDB	ADDTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
ADDTDQ												
ALLOC						X						
ALLOCA												
ALLOCD												
ASCLS												
ASCLSS												
CALQ												
CBLOCK												
CLASCG												
CLSDMP												
CORR												
DETLAG												
DTRNSF												
ERROR		X	X	X	X		X			X		
EXEC												
FORMC												
FORMQ												
FRETDB												
FRMPTB												
GENTDB												
GETCLS												
GETPTB												
GETTDB												
GRADF												
INIT												
INITR												
LAG												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	CLSDMP	CORR	DETLAG	DTRNSF	ERROR	EXEC	FORMC	FORMQ	FRETDB	FRMPTB	GENTDB	GETCLS
ADDTQ												
ALLOC												
ALLOCA												
ALLOCD												
ASCLS		X										
ASCLSS		X										
CALQ												
CLOCK				X		X						
CLASCG							X					
CLSDMP		X										
CORR						X						
DETLAG												
DTRNSF						X						
ERROR		X		X		X				X		
EXEC												
FORMC												
FORMQ												
FRETDB							X					
FRMPTB												
GENTDB												
GETCLS						X						
GETPTB												
GETTDB							X					
GRADF												
INIT							X					
INITR						X						
LAG						X						

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
ADTDQ												
ALLOC												
ALLOCA												
ALLOGD												
ASCLS												
ASCLSS												
CALO												
CBLOCK						X						
CLASCG											X	
CLSDMP												
CORR												
DETLAG												
DTRNSF												
ERROR						X		X				X
EXEC												
FORMC												
FORMQ												
PREIDB												
FRMPTB												
GENTDB												
GETCLS												
GETPTB												
GETTDB												
GRADF												
INIT												
INITR												
LAG												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	PLIST	PREPC	PTBOMP	PUTCLS	PUTPTB	REMCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
ADDIDQ												
ALLOC												
ALLOCA											X	
ALLOCD											X	
ASCLS												
ASCLSS												
CALQ												
CBLOCK												
CLASCG												
CLSDMP												
CORR												
DETLAG												
DTRNSF												
ERROR												
EXECT												
FORMC												
FORMQ												
FRETDB												
FRMPTB												
GENTDB												
GETCLS												
GETPTB												
GETTDB												
GRADE												
INIT												
INITR												
LAG												



SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRANS \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

INIT

ADDTQ	
ALLOC	
ALLOCA	
ALLOCD	
ASCLS	
ASCLSS	
CALQ	
CBLOCK	
CLASCG	
CLSDMP	
CORR	
DETLAG	
DTRNSF	
ERROR	X
EXECT	
FORMC	
FORMQ	
FRETDB	
FRMPTB	
GENTDB	
GETCLS	
GETPTB	
GETTDB	
GRADF	
INIT	
INITR	
LAG	

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMTE	UPDATE	WPTIE
LSTASK	X											
LSTRAK												
LSTSRC									X			
MLTCLS												
NEGUSE												
NEWCLS												
PBLOCK						X	X					
PLIST							X					
PREPC												
PTBDMP	X											
PUTCLS												
PUTPTB												
REMCLS												
RESINV												
RESUSE												
RUSER												
SCATSA												
SPLIT												
SRTCTP												
SRTTDB									X			
SVRUS1												
SVRUS2												
SYNC												
SYNCT						X						
TBLOCK							X					
TRACKD												
TRMTE	X											

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAH3 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	WRLTDS	WRUB	WRUDE	ADDTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
LSTASK												
LSTRAK												
LSTSRC												
MLTCLS												X
NEGUSE												
NEWCLS												
PBLOCK												
PLIST												
PREPC												
PTBDMP												
PUTCLS												
PUTPTB												
RENCLS												
RESINV												
RESUSE												
RUSER												
SCATSA												
SPLIT												
SRTCTP												
SRTTDB												
SVRUS1												
SVRUS2												
SYNC												
SYNCT												
TBLOCK												
TRACKD												
TRMTE												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	CLSDMP	CORR	DETLG	DTRNSF	ERROR	EXECT	FORMC	FORMQ	FRETDB	FRMPTB	GENTDB	GETCLS
LSTASK												
LSTRAK												
LSTSRC												
MLTCLS												
NEGUSE												
NEWCLS												
PBLOCK		X		X							X	
PLIST						X						
PREPC												
PTBDM												
PUTCLS				X								
PUTPTB										X		
REMCLS				X								
RESINV												
RESUSE						X						
RUSER												
SCATSA						X						
SPLIT												
SRTCTP				X								
SRTTDB												
SVRUS1												
SVRUS2												
SYNC						X						
SYNCT		X										
TBLOCK												
TRACKD						X						
TRMNT												

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	GETPTB	GETTDB	GRADF	INTR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
LSTASK												
LSTRAK												
LSTSRC												
MLTCLS												
NEGUSE												
NEWCLS												
PBLOCK												
PLIST												
PREPC												
PTBDMF												
PUTCLS												
PUTPTB												
RENCLS												
RESINV												
RESUSE												
RUSER												
SCATSA												
SPLIT												
SRCTP												
SRTTDB												
SVRUS1												
SVRUS2												
SYNC												
SYNCT												
TBLOCK												
TRACKD												
TRMTE												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUTCLS	PUTPIB	REMCLS	REMPIB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
LSTASK												
LSTRAK											X	
LSTSRC												
MLTCLS												
NEGUSE									X			
NEWCLS												X
PBLOCK												
PLIST												
PREPC												
PTBDM							X					
PUTCLS											X	
PUTPIB												
REMCLS												X
RESINV									X			
RESUSE										X		
RUSER												
SCATSA												
SPLIT												
SRTCTP												
SRTTDB												
SVRUS1												
SVRUS2									X			
SYNC										X		
SYNCT												
TBLOCK												
TRACKD												
TRANTE												



SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE OR ENTRY

USAGE SUMMARY

INIT

LSTASK	
LSTRAK	
LSTSRC	
MLTCLS	
NEGUSE	
NEWCLS	
PBLOCK	
PLIST	
PREPC	
PTBDM	
PUTCLS	
PUTPTB	
REMCLS	
RESINV	
RESUSE	
RUSER	
SCATSA	
SPLIT	
SRTCTP	
SRTTDB	
SVRUS1	
SVRUS2	
SYNC	
SYNCT	
TBLOCK	
TRACKD	X
TRMTE	

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRANS \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLUCK	TRACKD	TRMTE	UPDATE	WPTB
UPDATE												
WPTB												
WRLTDB									X			
WRUB												
WRUDB												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	WRLTDB	WRUB	WRUDB	ADDTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CAIG	CELOCK	CLASCG
UPDATE	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
WPTB	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
WRLTDB	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
WRUB	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
WRUDB	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*  
ROUTINE OR ENTRY  
USAGE SUMMARY

	CLSDMP	CORR	DETLG	DTRNSF	ERROR	EXECT	FORMC	FORMQ	FRETDB	FRMPTB	GENTDB	GETCLS
UPDATE						X						
WPTB												
WRLTDB												
WRUB												
WRUDB												

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLCK
UPDATE												
WPTB												
WRLTDB												
WRUB								X				
WRUDB								X				

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUTCLS	PUTPTB	REMCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
UPDATE												
WPTB												
WRLTDB												X
WRUB									X			
WRUDB									X			



SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

ROUTINE  
OR ENTRY

USAGE SUMMARY

INIT

UPDATE

WPTB

WRLTDB

WRUB

WRUDB

# SUBROUTINE CROSS-REFERENCE SUMMARY

ROUTINE  
OR ENTRY

## USAGE SUMMARY

	RDNAME	*BLOCK	NAME	NUMBER	CLOCK	BLKIN	BLOCK	INTRES	INTSOR	UPDRES	UPDSOR	GETRES
BLKIN					X							
INTRES					X							
INTSOR					X							
RDNAME					X							
UPDRES					X							
UPDSOR					X							

# SUBROUTINE CROSS REFERENCE SUMMARY

ROUTINE  
OR ENTRY

USAGE SUMMARY

	GETSOR	PUTRES	PUTSOR
BLKIN			
INTRES			
INTSOR			
RDNAM			
UPDRES			
UPDSOR			

### 3.6 Common Variable Definitions

The tables on the following pages define the meaning of each variable contained in each of the common blocks used by this program.

```

*****
*
* ROUTINES CONTROLLED BY
*
* IFLOW AND IDUMP
*
*****

```

```

*
* NOTE.- IFLOW CONTROLS THE PRINTING OF SUBROUTINE FLOW
*          MESSAGES IN KEY ROUTINES.
*          IF IFLOW=1, MESSAGE IS PRINTED AT ENTRY
*                TO SUBROUTINE.
*          IF IFLOW=0, MESSAGES ARE NOT PRINTED.
*
* IDUMP CONTROLS THE PRINTING OF DIAGNOSTIC MESSAGES
*          DURING THE EXECUTION OF KEY ROUTINES.
*          IF IDUMP=1, DIAGNOSTIC MESSAGES ARE PRINTED.
*          IF IDUMP=0, MESSAGES ARE NOT PRINTED.
*
*****

```

```

*****
*
* WORD      * ROUTINES AFFECTED
*
*****

```

```

*
* 1 * FORMQ
* 2 * ADDTDQ
* 3 * FORMC
* 5 * NEWCLS
* 6 * MLTCLS
* 7 * PREPC
* 8 * LSTASK
* 9 * LSTSRC,GENTDB,TRACKD
* 10 * DTRNSF
* 11 * SPLIT
* 13 * EXECT
* 14 * SYNCT
* 15 * SYNC
* 16 * CORR
* 17 * MERGE
* 18 * SCATSA
* 19 * RESUSE
* 20 * GRADF
* 21 * CLASCG
* 23 * RUSER
* 24 * CALQ
* 25 * RESINV
* 26 * FRMPTB
* 28 * FNDPTB
* 45 * MAIN
* 46 * TRACKD
* 50 * MAIN
*
*****

```

```

*****
*
*               C O M M O N   B L O C K   -   B L K S
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IBLOCK(4)    *  IBLOCK(1) - ADRESS OF FIRST PROCBLOCK
*                *  IBLOCK(2) - ADRESS OF FIRST TASK BLOCK.
*                *  IBLOCK(3) - ADRESS OF FIRST RESOURCE
*                *                UTILIZATION BLOCK. (RUB)
*                *  IBLOCK(4) - ADRESS OF FIRST RESOURCE
*                *                UTILIZATION DESCRIPTION
*                *                BLOCK (RUDB)
*  NBLOCK(4)    *  NBLOCK(1) - NUMBER OF PROCBLOCKS.
*                *  NBLOCK(2) - NUMBER OF TASK BLOCKS.
*                *  NBLOCK(3) - NUMBER OF RUBS.
*                *  NBLOCK(4) - NUMBER OF RUDBS.
*  LBLOCK(4)    *  LBLOCK(1) - LENGTH OF PROCBLOCK - 34.
*                *  LBLOCK(2) - LENGTH OF TASK BLOCK - 11.
*                *  LBLOCK(3) - LENGTH OF RUB - 9.
*                *  LBLOCK(4) - LENGTH OF RUDB - 8
*  IWORD( )     *  POOL OF STORAGE.
*
*****

```



```

*****
*
*           C O M M O N   B L O C K   -   C B L K
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NCOURS      *  NUMBER OF COURSES IN CURRENT RUN.
*  IADPB1(I)   *  ADRESS OF GRADUATION PROCBLOC FOR COURSE I.
*  ICTYPE(I)   *  COURSE TYPE.  1-CCTS
*              *  2-PMT
*  IPTYPE(I)   *  PERSONNEL TYPE. 1-PILOTS
*              *  2-COPILOTS
*              *  3-OSO
*              *  4-DSO
*  ICPRTY(I)   *  COURSE PRIORITY.
*  MAXCLS(I)   *  MAXIMUM CLASS SIZE.
*  IGINTR(I)   *  TIME INTERVAL BETWEEN GRADUATIONS.
*  ITGRD1(I)   *  TIME OF EARLIEST GRADUATION PERMITTED
*  ICGRAD(I)   *  TIME OF CURRENT (LATEST) GRADUATION.
*  IPGRAD(I)   *  TIME OF PREVIOUS (SMALLER TIME) GRADUATION.
*  NDXTD1(I)   *  POINTER TO FIRST ELEMENT IN TRAINING DEMAND
*              *  QUEUE (COMMON CTDQ) FOR COURSE I.
*  NDXTDL(I)   *  POINTER TO LAST ELEMENT IN TRAINING DEMAND
*              *  QUEUE (COMMON CTDQ) FOR COURSE I.
*  NDXND1(I)   *  ADRESS OF FIRST NODE FOR COURSE. (NOT USED).
*  FRCTN(I)    *  NUMBER OF STUDENTS ACCUMULATED FROM TRAINING
*              *  DEMANDS QUEUE THAT MUST BE PLACED IN CLASSES.
*
*              *
*              *
*              *  NOTE.- INDEX I REFERS TO COURSE NUMBER.
*
*****

```

```

*****
*
*          COMMON BLOCK - CCLS
*
*****
*
*  VARIABLE      *  DESCRIPTION
*
*****
*  NACLS          *  NUMBER OF CLASSES CURRENTLY ACTIVE.
*  IADRC(I)       *  ADDRESS OF CLASS I.
*  ICTME(I)       *  TIME FOR PROCESSING CLASS I.
*  ICPRT(I)       *  PRIORITY OF CLASS I.
*  IACTIVE(I)     *  CLASS STATUS. 0-ACTIVE.
*                  *  1-INACTIVE.
*  NOPB(I)        *  ADDRESS OF PROCBLOC ASSOCIATED WITH CLASS I.
*  IGID(I)        *  CREW IDENTIFICATION NUMBER.
*  ISORT( )       *  SORTED LIST OF INDICES FOR ACTIVE CLASSES.
*                  *  SORT IS ON TIME AND PRIORITY IN DESCENDING
*                  *  ORDER. (ISORT(1) CONTAINS INDEX OF CLASS
*                  *  WITH HIGHEST SCHEDULED EXECUTION TIME AND
*                  *  HIGHEST PRIORITY.)
*  ID(NER(I))     *  UNIQUE CLASS NUMBER ASSIGNED BY STEP 3
*                  *  OF TRAM.
*  LAGC(I)        *  DURATION OF CURRENT LAG FOR CLASS I.
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   C L A S S B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IFRSTC        *  POINTER TO FIRST ELEMENT IN LINKED LIST.
*  ILASTC        *  POINTER TO LAST ELEMENT IN LINKED LIST.
*  NXTFRE        *  POINTER TO NEXT UNUSED LOCATION.
*  LIMITC        *  DIMENSION LIMIT.
*  ICBSZE        *  NUMBER OF WORDS IN A CLASS BLOCK.
*  ICBSZ1        *  = ICBSZE - 1
*  ICLASS( )     *  LINKED LIST STORAGE AREA FOR CLASS BLOCKS.
*                *  SEE DESCRIPTION OF CLASS BLOCKS (FIG 8.1)
*                *  FOR DEFINITION OF PARAMETERS.
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   C L S B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  NDXCLS      *  ADRESS OF CURRENTLY ACTIVE CLASS.
*  NUMCRS      *  COURSE NUMBER.
*  NOSTDS      *  NUMBER OF STUDENTS.
*  IPRTYC      *  CLASS PRIORITY.
*  ICLSTM      *  TIME OF NEXT SCHEDULED EVENT FOR CLASS.
*  NPROCB      *  ADRESS OF PROCBLOC FOR CLASS.
*  ISTATS      *  CLASS STATUS.  0-ACTIVE.
*              *  1-INACTIVE.
*  IPREDT      *  ADRESS OF NEXT PREDETERMINED TRANSFER
*              *  BLOCK FOR CLASS.
*  IDGRAD      *  CREW IDENTIFICATION NUMBER.
*  NUMGRD      *  ADRESS OF GRADUATION PROCBLOC.
*  IDTEGR      *  GRADUATION DATE.
*  IC          *  POINTER TO CLASS IN LIST OF CURRENT CLASSES.
*  IDIDR      *  UNIQUE CLASS NUMBER.
*  LAGT        *  DURATION OF CURRENT LAG (IN C.U.S)
*  LRSON       *  REASON FOR CURRENT LAG.  1-RESOURCE MISSING.
*              *  6-SYNCHRONIZATION FAILURE
*              *  7-CORRELATION FAILURE.
*  ICLSAD      *  ADRESS OF CLASS IN MASS STORAGE.
*  ICRSPY      *  COURSE PRIORITY.
*  ISORCN      *  SOURCE NUMBER FOR CLASS.
*  LAGTOT      *  TOTAL TIME CLASS HAS BEEN LAGGED.
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   C L S T
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  INDXC(I)      *  POINTERS TO CLASSES (IN COMMON CCLS) THAT ARE
*                  *  CURRENTLY IN PROCBLOCS LINKED TOGETHER BY A
*                  *  SYNCHRONIZATION OR CORRELATION LOOP.
*  IACLS(I)      *  WORK AREA. POINTERS TO CLASSES ASSOCIATED WITH
*                  *  A SPECIFIC PROCBLOCK.
*  IPBLK(I)      *  WORK AREA. COURSE NUMBER OF ITH COURSE IN
*                  *  SYNC. LOOP.
*  NSTUDS(I)     *  WORK AREA. NUMBER OF STUDENTS IN ITH COURSE
*                  *  OF SYNCH. LOOP.
*  NOCLS         *  NUMBER OF CLASSES IN INDXC.
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   C O N T R L
*
*****
*
*  VARIABLE      *      D E S C R I P T I O N
*
*****
*  ICLK          *  CLOCK TIME AT WHICH LAST UPDATE OF SOURCES
*                  *  AND RESOURCES WAS DONE.
*
*****

```



```

*****
*
*          C O M M O N   B L O C K   -   C R S G R P
*
*****
*          *
*  VARIABLE *  D E S C R I P T I O N
*          *
*****
*          *
*  NCGRPS   *  NUMBER OF COURSE GROUPS.
*  IUPTCG   *  NOT USED.
*  NUMCRU   *  NEXT AVAILABLE CREW NUMBER - 1.
*  NCING(I) *  NUMBER OF COURSES IN GROUP I.
*  ICING(J,I) *  NUMBER OF JTH COURSE IN GROUP I.
*          *
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   C T D Q
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  I1FREE        *  POINTER TO NEXT AVAILABLE LOCATION
*  LIMIT1        *  DIMENSION LIMIT FOR ARRAY ICORE.
*  ICORE( )      *  LINKED LIST STORAGE FOR TRAINING DEMANDS
*                *  (SEE FIGURE 8.2   FOR DESCRIPTION OF
*                *  TRAINING DEMAND BLOCKS)
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   E C B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  ITIMES        *  START TIME OF SIMULATION.
*  ITIMEE        *  END TIME OF SIMULATION.
*  ITIMEC        *  CURRENT TIME OF SIMULATION.
*  NXTBRK        *  TIME OF NEXT SCHEDULED EVENT.
*  MINGRD        *  TIME OF EARLIEST CURRENT GRADUATION.
*  ITEVNT        *  TIME OF LATEST PREVIOUS GRADUATION.
*  NCRSES        *  NUMBER OF COURSES.
*  NCLSES        *  TOTAL NUMBER OF CLASSES CURRENTLY IN SYSTEM.
*  IDRS          *  DEMAND RECORD STATUS. 1- FIRST TIME
*                *  2- NORMAL
*                *  3- E.O.F.
*  NOTDRS        *  NUMBER OF TRAINING DEMAND RECORDS READ.
*  ITRNRU        *  FORTRAN UNIT NUMBER FOR READING TRAINING
*                *  DEMAND RECORDS.
*  ITRANW        *  NOT USED
*  IFAIL         *  IF =0, TASK EXECUTION SUCCEEDED.
*                *  IF =1, TASK EXECUTION FAILED.
*  KEOF          *  NOT USED
*  IFAIL1        *  NOT USED
*  IFAIL2        *  NOT USED
*  IOPTF         *  PROCESSING OPTION IN CASE OF RESOURCE ALLOCATION
*                *  FAILURE. 0- STOP
*                *  1- IGNORE AND CONTINUE
*                *  2- LAG
*  IOPTF1        *  NOT USED.
*  IOPTF2        *  NOT USED
*  MAXLAG        *  MAXIMUM LENGTH OF TIME A CLASS CAN BE LAGGED
*                *  BEFORE IT STARTS EXECUTING THE EXTRAS TASKS
*                *  IF IT CANNOT BE CURRELATED IMMEDIATELY.
*  IEXTRA        *  IF SET TO 1 (BY CORR) INDICATES THAT THE
*                *  EXTRAS TASKS SHOULD BE EXECUTED.
*  IFLOW(50)     *  SWITCHES TO CONTROL PRINTING OF PROGRAM FLOW
*                *  INFORMATION. 0-NO PRINT, 1-PRINT.
*                *  (SEE TABLE 7.1 FOR DETAILS)
*  IDUMP(50)     *  SWITCHES TO CONTROL PRINTING OF DIAGNOSTICS
*                *  DURING PROGRAM EXECUTION. 0-NO PRINT, 1-PRINT.
*                *  (SEE TABLE 7.1 FOR DETAILS)
*  ICORSE( )     *  NOT USED.
*
*****

```

```

*****
*
*          COMMON BLOCK - PB
*
*****
*
*  VARIABLE      *  DESCRIPTION
*
*****
*  IPROC B      *  PROCBLOCK NUMBER.
*  IBTYPE       *  1
*  IDURAT       *  DURATION OF PROCBLOCK. (IN CALENDAR UNITS)
*  IPRTY        *  PRIORITY OF PROCBLOCK.
*  ISYNCT       *  COURSE NUMBER.
*  IDSYNB       *  ADDRESS OF NEXT PROCBLOCK IN SYNCHRONIZATION
*                *  OR CORRELATION LOOP. 0 IF PROCBLOCK IS NOT
*                *  SYNCHRONIZED OR CORRELATED WITH ANY OTHER
*                *  PROCBLOCKS.
*  NLBRNC       *  NUMBER OF LEFT BRANCHES IN PROCBLOCK.
*  LBRNCH(1,I)  *  ADDRESS OF NEXT PROCBLOCK ALONG BRANCH I.
*  LBRNCH(2,I)  *  TRANSFER PRIORITY OF ITH BRANCH.
*  LBRNCH(3,I)  *  TRANSFER PROPORTION ALONG BRANCH I.
*  NTASKS       *  NUMBER OF TASKS.
*  ITASK(J)     *  ADDRESS OF JTH TASK.
*  NRBRNC       *  NUMBER OF RIGHT BRANCHES IN PROCBLOCK.
*  IRBRNC(K)    *  ADDRESS OF NEXT PROCBLOCK ALONG THE KTH BRANCH.
*                *  (IN A LEFT TO RIGHT DIRECTION)
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   P T B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  PROP(5)       *  PROPORTIONS FOR 5 LEFT BRANCHES OF
*                 *  PROCBLOC.
*  NEXTPT(5)     *  ADRESSES OF NEXT PTBS ALONG EACH OF THE
*                 *  5 BRANCHES.
*  IPROP(10)     *  FIRST 10 WORDS OF ACTIVE PTB.
*                 *  IPROP(1-5) CONTAIN THE NUMBER OF
*                 *  STUDENTS TO BE SENT ALONG EACH BRANCH.
*                 *  IPROP(6-10) CONTAIN THE ADRESSES OF THE
*                 *  NEXT PTBS ALONG EACH OF THE FIVE BRANCHES.
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   P T B C
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  I1PTB         *  POINTER TO FIRST PREDETERMINED TRANSFER BLOCK.
*  ILPTB         *  POINTER TO LAST PREDETERMINED TRANSFER BLOCK.
*  NXTFPT        *  POINTER TO NEXT AVAILABLE LOCATION FOR STORING
*                 *  PTBS.
*  LIMPTB        *  DIMENSION LIMIT FOR PTB STORAGE ARRAY.
*  ISZEPT        *  SIZE OF PTB RECORD (11)
*  ISZ1PT        *  = ISZEPT - 1 = 10
*  ICORPT( )     *  LINKED LIST STORAGE AREA FOR PTBS
*                 *  (FOR DETAILED DESCRIPTION OF PREDETERMINED
*                 *  TRANSFER BLOCK SEE COMMON BLOCK PTB).
*
*****

```



```

*****
*
*               C O M M O N   B L O C K   -   R E S
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NRESCR        *  NUMBER OF RESOURCES CURRENTLY IN LOCAL
*                 *  STORAGE.
*  IBUCKET(I)    *  BUCKET SIZE FOR RESOURCE I.
*  INCORE(I)     *  IF =0 RESOURCE I IS NOT IN LOCAL STORAGE.
*                 *  =1 RESOURCE I IS IN LOCAL STORAGE
*  LOTIM1(I)     *  START TIME FOR LOCAL INVENTORY OF RESOURCE I.
*  LOTIM2(I)     *  END TIME FOR LOCAL INVENTORY OF RESOURCE I.
*  NBUCKET(I)    *  NUMBER OF BUCKETS OF RESOURCE I IN LOCAL
*                 *  STORAGE.
*  IT1(I)        *  BUCKET NUMBER CORRESPONDING TO LITIM1.(ASSUMES
*                 *  RES. INVENTORY STARTS AT TIME=1)
*  IT2(I)        *  BUCKET NUMBER CORRESPONDING TO LITIM2.
*                 *  (FOR ITH RESOURCE)
*  NXT           *  NEXT AVAILABLE LOCATION IN ARRAY INVRES.
*  LIMNXT        *  DIMENSION LIMIT FOR ARRAY INVRES.
*  IA1           *  THEORETICAL NUMBER OF BUCKET CORRESPONDING
*                 *  TO ITIME1. (WORK VARIABLE FOR CURR. RESOURCE)
*  IA2           *  THEORETICAL NUMBER OF BUCKET CORRESPONDING
*                 *  TO ITIME2. (WORK VARIABLE FOR CURRENT RESOURCE).
*  INDX1         *  POINTER TO ELEMENT IN ARRAY INVRES THAT
*                 *  CORRESPONDS TO THE 'HIGH-TIME' BUCKET OF THE
*                 *  DESIRED RES. INVENTORY.
*  INDX2         *  POINTER TO ELEMENT IN ARRAY INVRES THAT
*                 *  CORRESPONDS TO THE 'LOW-TIME' BUCKET OF THE
*                 *  DESIRED RESOURCE INVENTORY.
*  NBI           *  NUMBER OF RESOURCE BUCKETS REQUIRED TO COVER
*                 *  THE ACTIVE PROCBLOC.
*  LIMRES        *  DIMENSION LIMIT FOR ARRAY INVRES.
*  INVRES( )     *  TEMPORARY STORAGE AREA FOR INVENTORIES OF
*                 *  CURRENTLY ACTIVE RESOURCES.
*  LIMNS         *  DIMENSION LIMIT FOR ARRAYS IADI1,IADI2 AND IADS1.
*  LIMIS        *  DIMENSION LIMIT FOR ARRAY IAUSED.
*  NSAVE         *  NUMBER OF RESOURCES WHOSE CONSUMPTION HAS BEEN
*                 *  STORED IN ARRAY IAUSED.
*  ISAVE         *  NUMBER OF BUCKETS USED TO STORE TEMPORARY
*                 *  UPDATE OF RESOURCE UTILIZATION.
*  IADI1(J)      *  SAVED VALUE OF INDX1. (POINTER TO 'HIGH-TIME'
*                 *  BUCKET IN INVRES).
*  IADI2(J)      *  SAVED VALUE OF INDX2. (POINTER TO 'LOW-TIME'
*                 *  BUCKET IN INVRES).
*  IADS1(J)      *  POINTER TO FIRST ELEMENT SAVED IN IAUSED.
*                 *  (CORRESPONDS TO IADI1)
*                 *  NOTE.- J VARIES FROM 1 TO NSAVE.
*  IAUSED(K)     *  SAVED RESOURCE INVENTORIES.
*                 *  NOTE.- K VARIES FROM 1 TO ISAVE.
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   R L T D B C
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  NTDBRL      *  NUMBER OF RIGHT-TO-LEFT TRACK DESCRIPTOR
*                *  BLOCKS.
*  ITRK1(J)    *  POINTER TO FIRST TRACK DESCRIPTOR BLOCK
*                *  FOR COURSE J. (NOTE.-A DUMMY ENTRY IS
*                *  MADE FOR THE LAST+1 COURSE)
*  NXTNDA(I)   *  POINTER TO NEXT NODE (TDB) ALONG THIS
*                *  TRACK. (IN A RIGHT TO LEFT DIRECTION).
*  ITDURT(I)   *  TIME DURATION BETWEEN SOURCE AND NODE I.
*                *  (INCLUDING THE DURATION OF THE NODE PROCBLOC).
*  CUMPCT(I)   *  CUMULATIVE TRANSFER PROPORTION FROM SOURCE
*                *  TO CURRENT NODE. OBTAINED BY MULTIPLYING
*                *  THE SPECIFIED PROPORTIONS ALONG THE TRACK)
*  CUMPTY(I)   *  CUMULATIVE PRIORITY OF TRACK AT THIS NODE.
*                *  (=CURRENT PRIORITY + PREVIOUS PRIORITY / 100.,
*                *  APPLIED RECURSIVELY.)
*  NLFTB(I)    *  NUMBER OF LEFT BRANCH IN PROCBLOC.
*  NSRCE(I)    *  POINTER TO SOURCE DESCRIPTION. (COMMON SORDSC)
*  NUMBLK(I)   *  NUMBER OF PROCBLOC ASSOCIATED WITH THIS NODE.
*  ITDBST( )   *  LIST OF SORTED POINTERS TO TDBS.
*                *  (SORT IS IN DESCENDING ORDER ON PRIORITY BY
*                *  COURSE).
*  NUMSTA(I)   *  NUMBER OF STUDENTS ASSIGNED TO THIS NODE.
*                *  (CALCULATED FOR EACH CLASS BY SCATSA).
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   R U B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  IBLOCN      *  RUB NUMBER (PRIMARY)
*  IBLKT       *  3
*  NRUDBS      *  NUMBER OF RUDBS USED BY PRIMARY RUB.
*  IARUDB(I)   *  ADDRESS OF ITH RUDB. (PRIMARY)
*  JBLOCN      *  NUMBER OF SECONDARY RUB.
*  JBLKT       *  3
*  MRUDBS      *  NUMBER OF RUDBS USED BY SECONDARY RUB.
*  JARUDB(I)   *  ADDRESS OF ITH SECONDARY RUDB.
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   R U D B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  INTBN          *  NUMBER OF CURRENT PRIMARY RUDB.
*  IBTYPE         *  4
*  IRESNO         *  NUMBER OF RESOURCE DESCRIBED BY RUDB.
*  IRUGF          *  NUMBER OF RESOURCE UTILIZATION GROUPING FUNCTION.
*                 *  1.- CLASS.
*                 *  3.- INDIVIDUAL.
*  IRUTF          *  NUMBER OF RESOURCE UTILIZATION TIMING FUNCTION.
*                 *  2.- UNIFORM
*                 *  3.- ARBITRARY.
*  NXTRUB         *  ADDRESS OF SECONDARY RUB.
*  IALTR          *  ADDRESS OF ALTERNATE RUDB.
*  ICONSU         *  UNITS OF CONSUMPTION PER UNIT USER.
*  JNTBN          *  NUMBER OF CURRENT SECONDARY RUDB.
*  JBTYPE         *  4
*  JRESNO         *  NUMBER OF RESOURCE DESCRIBED BY RUDB.
*  JRUGF          *  NUMBER OF RESOURCE UTILIZATION GROUPING FUNCTION.
*                 *  1.- CLASS
*                 *  2.- QUANTITY OF PRIMARY RESOURCE CONSUMED.
*                 *  3.- INDIVIDUAL
*  JRUTF          *  NUMBER OF RESOURCE UTILIZATION TIMING FUNCTION.
*                 *  2.- UNIFORM
*                 *  3.- ARBITRARY.
*  MXTRUB         *  NOT USED
*  JALTR          *  ADDRESS OF ALTERNATE RUDB.
*  JCONSU         *  UNITS OF CONSUMPTION PER UNIT USER.
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   S O R D S C
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NOSRCS        *  NUMBER OF SOURCES (AS DEFINED BY COURSE
*                  *  TRACKS)
*  ISRCPB(I)     *  ADDRESS OF PROCBLOC (AT SOURCE).
*  ISTASK(I)     *  ADDRESS OF GETSOURCE TASK.
*  ISRRUB(I)     *  ADDRESS OF RUB.
*  ISRUB(I)      *  ADDRESS OF RUDB.
*  ISORNN(I)     *  SOURCE NUMBER.
*
*              *  NOTE.- I LOOPS OVER ALL TRACKS.
*
*****

```



```

*****
*
*           C O M M O N   B L O C K   -   T B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*
*  IBLKN          *  NUMBER OF TASK BLOCK.
*  IBLKT1         *  2
*  ITSKFN         *  TASK FUNCTION NUMBER.
*  ITSKPT         *  TASK TYPE.
*  IARUB          *  ADDRESS OF ASSOCIATED RUB.
*  NPARMS         *  NUMBER OF PARAMETERS ASSOCIATED WITH TASK.
*  IPARM(I)       *  ITH PARAMETER
*  ITIME1         *  START TIME FOR THIS TASK
*  ITIME2         *  END TIME FOR THIS TASK.
*  LITIM1         *  EARLIEST START TIME FOR ALL TASKS IN THE
*                  *  CURRENTLY ACTIVE TASK LIST.
*  LITIM2         *  LATEST START TIME FOR ALL TASKS IN THE
*                  *  CURRENTLY ACTIVE TASK LIST.
*  ICLSIZE        *  CLASS SIZE ASSOCIATED WITH THIS TASK.
*
*****

```



```

*****
*
*           C O M M O N   B L O C K   -   T D R
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  ITDATE        *  TIME OF TRAINING DEMAND
*  STUDNO        *  NUMBER OF STUDENTS DEMANDED.
*  ITTYPE        *  TRAINEE TYPE. 1-PILOT
*                *                2-COPILOT
*                *                3-OSO
*                *                4-DSO
*  ICRSN         *  COURSE NUMBER.
*  IDGRAD        *  NOT USED. (GRADUATION ID GENERATED BY STEP 2).
*  IDTYPE        *  DEMAND TYPE. 1- CCTS DUE TO DELIVERIES.
*                *                2- CCTS DUE TO ATTRITION.
*                *                3- PMT
*
*****

```

```

*****
*
*          C O M M O N   B L O C K   -   T L I S T
*
*****
*
*          *
*  VARIABLE *   D E S C R I P T I O N
*          *
*****
*          *
* NSYNCT    * NUMBER OF SYNC TASKS IN LIST.
* NTSKS     * NUMBER OF TASKS IN LIST.
* IDSTSK(I) * TASK ADDRESS.
* ICLSID(I) * POINTER TO CLASS IN LIST OF CURRENT CLASSES.
* ICOMID(I) * IF =0, THEN TASK I IS A CLASS TASK.
*          * IF =1, THEN TASK I IS A COMMON TASK.
* KTIME1(I) * START TIME FOR TASK I.
* KTIME2(I) * END TIME FOR TASK I. (TIMES ARE ONLY
*          * APPLICABLE TO RESOURCE UTILIZATION TASKS)
* KLASZE(I) * CLASS SIZE TO BE USED FOR TASK I.
* LTIME1    * MINIMUM OF KTIME1 ENTRIES.
* LTIME2    * MAXIMUM OF KTIME2 ENTRIES.
*          *
*****

```

```

*****
*
*           C O M M O N   B L O C K   -   W O R K B
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  IPBLOC(34) *  WORK AREA FOR STORING A PROCBLOC.
*              *  (SEE FIG 8.3   FOR DETAILED DESCRIPTION
*              *  OF PROCBLOC).
*  IWTASK(12) *  WORK AREA FOR STORING A TASK BLOCK.
*              *  (SEE FIG 8.4   FOR DETAILED DESCRIPTION
*              *  OF TASK BLOCK).
*  IADRSB( )  *  NOT USED.
*
*****

```

```

*****
*
*           C O M M O N   B L O C K   - W R K A
*
*****
*
*  VARIABLE      *  D E S C R I P T I O N
*
*****
*  NUMTR         *  NUMBER OF CURRENT TRAINING DEMANDS FOR
*                 *  COURSE IN PROCESS.
*  JDATE(I)      *  TIME OF ITH TRAINING DEMAND.
*  STUDSN(I)     *  NUMBER OF STUDENTS IN ITH TRAINING DEMAND.
*  JID(I)        *  NOT USED.(CREW # GENERATED BY STEP 2)
*  JTTYPE(I)     *  TRAINEE TYPE.  1-PILOTS
*                 *  2-COPILOTS
*                 *  3-OSO
*                 *  4-OSO
*  JDTYPE(I)     *  DEMAND TYPE. 1-CCTS DUE TO DELIVERY.
*                 *  2-CCTS DUE TO ATTRITION
*                 *  3-PMT
*                 *  4-ROUND OFF GENERATED BY PROGRAM
*  LIMTR         *  DIMENSION LIMIT FOR TRAINING DEMANDS FOR
*                 *  ONE COURSE.
*
*****

```

### 3.7 Internal Data Block Description

The tables on the following pages define the contents of each of the data blocks used by the Phase 3 TRAM program.

```

*****
*
*      C L A S S   B L O C K
*
*****
*
*      *
* WORD      *   D E S C R I P T I O N
*      *
*****
*      *
*      1      *   COURSE NUMBER.
*      2      *   NUMBER OF STUDENTS IN CLASS.
*      3      *   CLASS PRIORITY.
*      4      *   TIME OF NEXT SCHEDULED EVENT FOR CLASS.
*      5      *   ADDRESS OF NEXT ACTIVE PROCBLOC FOR CLASS.
*      6      *   CLASS STATUS.  0-ACTIVE.
*              *   1-INACTIVE.
*      7      *   ADDRESS OF NEXT PREDETERMINED TRANSFER BLOCK
*              *   FOR CLASS.
*      8      *   CREW IDENTIFICATION NUMBER.
*      9      *   ADDRESS OF GRADUATION PROCBLOC.
*     10      *   GRADUATION DATE.
*     11      *   POINTER TO CLASS IN LIST OF CURRENT CLASSES.
*     12      *   UNIQUE CLASS NUMBER.
*     13      *   DURATION OF CURRENT LAG.
*     14      *   REASON FOR CURRENT LAG.  1-RESOURCE MISSING.
*              *   6-SYNC. FAILURE.
*              *   7-CORR. FAILURE.
*     15      *   ADDRESS OF CLASS IN MASS STORAGE.
*     16      *   COURSE PRIORITY.
*     17      *   SOURCE NUMBER FOR CLASS. (NOT USED).
*     18      *   TOTAL TIME CLASS HAS BEEN LAGGED.
*     30      *   LINK TO NEXT CLASS BLOCK.
*
*****

```



```

*****
*
*   T R A I N I N G   D E M A N D   B L O C K
*
*****
*   *
* WORD *   D E S C R I P T I O N
*   *
*****
*   *
*   1 *   NUMBER OF STUDENTS.
*   2 *   DEMAND TIME
*   3 *   0
*   4 *   TRAINEE TYPE
*   5 *   DEMAND TYPE
*   6 *   POINTER TO NEXT DEMAND FOR COURSE.
*   *
*   *   NOTE.- NDXTD1(I) IN COMMON CBLK CONTAINS A
*   *   POINTER TO THE FIRST TRAINING DEMAND
*   *   FOR THE ITH COURSE.
*   *   NDXTDL(I) IN COMMON CBLK CONTAINS A
*   *   POINTER TO THE LAST TRAINING DEMAND
*   *   FOR THE ITH COURSE.
*   *
*****

```

```

*****
*
*      P R O C E S S I N G   B L O C K
*
*****
*
*  WORD      *  D E S C R I P T I O N
*
*****
*
*  1  *  INTERNAL BLOCK NUMBER.
*  2  *  BLOCK TYPE. (1-PROC6LOC)
*  3  *  DURATION.
*  4  *  BLOCK PRIORITY
*  5  *  SYNCHRONIZATION TYPE.
*  6  *  NUMBER OF PROC6LOC SYNCHRONIZED WITH.
*  7  *  NUMBER OF LEFT BRANCHES.
*  8  *  LEFT BRANCH POINTER 1
*  9  *  PRIORITY 1
* 10  *  PERCENTAGE 1
* 11  *  LEFT BRANCH POINTER 2
* 12  *  PRIORITY 2
* 13  *  PERCENTAGE 2
* 14  *  LEFT BRANCH POINTER 3
* 15  *  PRIORITY 3
* 16  *  PERCENTAGE 3
* 17  *  LEFT BRANCH POINTER 4
* 18  *  PRIORITY 4
* 19  *  PERCENTAGE 4
* 20  *  LEFT BRANCH POINTER 5
* 21  *  PRIORITY 5
* 22  *  PERCENTAGE 5
* 23  *  NUMBER OF TASKS
* 24  *  POINTER TO TASK 1
* 25  *  POINTER TO TASK 2
* 26  *  POINTER TO TASK 3
* 27  *  POINTER TO TASK 4
* 28  *  POINTER TO TASK 5
* 29  *  NUMBER OF RIGHT BRANCHES
* 30  *  RIGHT BRANCH POINTER 1
* 31  *  RIGHT BRANCH POINTER 2
* 32  *  RIGHT BRANCH POINTER 3
* 33  *  RIGHT BRANCH POINTER 4
* 34  *  RIGHT BRANCH POINTER 5
*
*****

```

```

*****
*
*           T A S K   B L O C K
*
*****
*
*  WORD      *  D E S C R I P T I O N
*
*****
*
*  1  *  I N T E R N A L   B L O C K   N U M B E R .
*  2  *  B L O C K   T Y P E . (2-TASK BLOCK)
*  3  *  T A S K   F U N C T I O N   N U M B E R . (NUMBER OF ROUTINE INVOKED)
*  4  *  T A S K   T Y P E .  1-NORMAL
*      *  2-EXTRAS
*  5  *  P O I N T E R   T O   R E S O U R C E   U T I L I Z A T I O N   B L O C K .
*  6  *  N U M B E R   O F   P A R A M E T E R S
*  7  *  P A R A M E T E R   1
*  8  *  P A R A M E T E R   2
*  9  *  P A R A M E T E R   3
* 10  *  P A R A M E T E R   4
* 11  *  P A R A M E T E R   5
*
*****

```

```

*****
*
*   RESOURCE UTILIZATION BLOCK
*
*****
*   *
* WORD * DESCRIPTION
*   *
*****
*   *
* 1 * INTERNAL BLOCK NUMBER.
* 2 * BLOCK TYPE. (3-RUB)
* 3 * NUMBER OF RESOURCE UTILIZATION DESCRIPTION BLOCKS.
* 4 * POINTER TO RUDB 1
* 5 * POINTER TO RUDB 2
* . *
* 9 * POINTER TO RUDB 6
*   *
*****

```

```

*****
*
*          RESOURCE UTILIZATION DESCRIPTION BLOCK
*
*****
*          *
* WORD    *  D E S C R I P T I O N
*          *
*****
*          *
*      1  *  INTERNAL BLOCK NUMBER
*      2  *  BLOCK TYPE. (4-RUDB)
*      3  *  RESOURCE NUMBER
*      4  *  RESOURCE UTILIZATION GROUPING FUNCTION NUMBER.
*      5  *  RESOURCE UTILIZATION TIMING FUNCTION NUMBER.
*      6  *  POINTER TO NEXT RUB. (FOR COMPOSITE RESOURCES)
*      7  *  POINTER TO ALTERNATE RUDB.
*      8  *  UNITS OF CONSUMPTION / UNIT USER.
*          *
*****

```

3.8

Common Variable Cross Reference Table

The tables on the following pages show how each subroutine uses each common variable. The subroutine names are printed across the top of the table, and the variable names down the left side.



## CROSS REFERENCE USAGE CODES

### A ARGUMENT

THE SYMBOL IS A VARIABLE OR FUNCTION NAME WHICH APPEARS IN AN ARGUMENT LIST OF A CALL, SUBROUTINE, FUNCTION, OR ENTRY STATEMENT.

### D DATA INITIALIZATION

THE SYMBOL IS A VARIABLE WHICH IS INITIALIZED IN A DATA OR TYPE SPECIFICATION STATEMENT SUCH AS A COMPLEX SPECIFICATION STATEMENT.

### F FETCH A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE MOST RECENTLY ASSIGNED VALUE IS ACCESSED BUT NOT CHANGED.
2. FUNCTION NAME OR ARGUMENT OF A FUNCTION WHICH APPEARS ON THE RIGHT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT OR APPEARS IN AN IF STATEMENT TEST.
3. DUMMY ARGUMENT IN A STATEMENT FUNCTION DEFINITION.

### S STORE A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE VALUE IS REPLACED BY ANOTHER VALUE.
2. FUNCTION NAME WHICH APPEARS ON THE LEFT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT.
3. NAME OF A STATEMENT FUNCTION IN THE DEFINITION OF THAT FUNCTION.

### C COMMON

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A COMMON STATEMENT OR IS THE NAME OF A LABELED COMMON BLOCK.

### E EQUIVALENCE

THE SYMBOL IS A VARIABLE WHICH APPEARS IN AN EQUIVALENCE STATEMENT.

### T TYPE SPECIFICATION

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A :

1. TYPE SPECIFICATION STATEMENT AND IS NOT INITIALIZED IN THAT STATEMENT.
2. DIMENSION OR EXTERNAL STATEMENT.

### N ENTRY POINT

THE SYMBOL IS AN ENTRY POINT DEFINED BY AN ENTRY STATEMENT IN A SUBROUTINE OR FUNCTION.

### X EXTERNAL REFERENCE

THE SYMBOL IS A SUBROUTINE OR ENTRY NAME WHICH APPEARS IN A CALL STATEMENT.

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAN3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	MAIN	SRCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMTE	UPDATE	WPTB
CBLK	C					C			C			
CCLS	C					C	C					
CLASSB	C					C						
CLSB	C					C						
CRSGRP												
CTDQ												
CUMPT												
CUMPTY			A F T						F C			
ECB	C					C			A F C			
FRCTN	C					C			C			
IACTIVE	F C					C	SC		C			
IAD11				SC							C	
IAD12				SC							C	
IADPB1	C					C			A C			
IADRC	F C					F C	F C					
IADRSB												
IADS1				SC							C	
IALTR												
IARUB												
IARUB							C					
IAUSED				SC							C	
IA1					FSC						C	
IA2				C		C					C	
IBLKN				C		C					C	
IBLKT							A C					
IBLKT1												
IBLOCK							C					
								A F T				

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	WRLTDB	WRUR	WRUDS	ADTDE	ALLEC	ALLOCA	ALLECO	ASCLS	ASCLES	CALQ	CELCK	CLASCG
CBLK	CB				C								C
CCLS	CB								C				
CLASSB	CB												
CLSB	CB												
CRSGRP	CS												C
CTDQ	CB				C								
CUMPT	R	F C				C		F C					
CUMPTY	R	F C				C		C					
ECB	CB				C								C
FRCIN	R				C								FSC
IACTIVE	I								C				
IAD11	I												
IAD12	I												
IADP61	I				C								C
IADRC	I												
IADRSB	I								C				
IADSL	I												
IALTR	I												
IARUB	I												
IARUDB	I												
IAUSED	I												
IA1	I												
IA2	I												
IBLKN	I												
IBLKT	I												
IBLKT1	I												
IBLOCK	I		A F	T	A F	T						A F	T

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	CLSDMP	CORR	DETLG	DTANSE	ERROR	EXECT	FORMC	FORMQ	FRETD6	FRMPTB	GENID6	GETCLS
CBLK												
CCLS		C										
CLASSB	C	C										
CLSB		C										
CRSGRP												
CTDQ												
CUMPT												
CUMPTY												
ECB		C										
FRCTN												
IACTVE		C										
IAD11												
IAD12												
IADPB1												
IADRC												
IADRSB												
IADSL												
IALTR												
IARUB												
IARUDB												
IAUSED												
IA1												
IA2												
IBLKN												
IBLKT												
IBLKT1												
IBLOCK												

# CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	GEIPTB	GETTDB	GRADDF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
CBLK	C								C		C	
CCLS					C	C						
CLASSB					C						C	
CLSB						C						
CRSGRP								C				
CTDQ		C										
CUMPT							C					
CUMPTY							FC					
ECB			C			C			C		C	
FRCN		C							C		C	
IACTIVE			FSC		C	C						
IAD11				C						FC		
IAD12				C						FC		
IADPB1	C		C					A F	C		FC	
IADRC					FC	C						
IADRSB								A FS				
IADS1				C						FC		
IALTR												
IARUB						C						
IARUDB												
IAUSED				C						FC		
IA1				C								
IA2				C						C		
IBLKN				C						C		
IBLKT					A	C						
IBLKT1												
IBLOCK						C						A F



CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUTCLS	PUTPTB	REMCLS	REMPBTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
CBLK												
CCLS		C										
CLASSB		C		C		C						C
CLSB											C	C
CRSGRP												
CTDQ												
CUMPTCT												
CUMPTY												
ECB		C				C		C		C		C
FRCTN												
IACTIVE		FSC										
IAD11								C	C			
IAD12								C	C			
IADPB1												
IADRC		FSC										
IADRSB												
IADSI								C	C			
IALTR									F C			
IARUB									A C			
IARUDB									A SC			
IAUSED								C	C			
IA1								FSC	C			
IA2								FSC	C			
IBLKN					A S				A C			
IBLKT									C			
IBLKT1									C			
IBLOCK									C			



CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

		INIT
CBLK	CB	C
CCLS	CB	C
CLASSB	CB	C
CLSB	CB	
CRSGRP	CB	C
CTOQ	CB	C
CUMPT	R	
CUMPTY	R	
ECB	CB	C
FRCTN	R	SC
IACTIVE	I	SC
IAD11	I	C
IAD12	I	C
IADPB1	I	C
IADRC	I	SC
IADRSB	I	
IADSL	I	C
IALTR	I	
IARUB	I	
IARUDB	I	
IAUSED	I	C
IA1	I	C
IA2	I	C
IBLKH	I	
IBLKT	I	
IBLKT1	I	
IBLOCK	I	

# CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

## USAGE SUMMARY

	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMTE	UPDATE	WPTB
IBLOCN	I											
IBTYPE	I											
IBUCKT	I											
IC	I											
ICBSZE	I											
ICBSZ1	I											
ICGRAD	I											
ICING	I											
ICLASS	I											
ICLSAD	I											
ICLSID	I											
ICLSTM	I											
ICLSZE	I											
ICOMID	I											
ICONSU	I											
ICORE	I											
ICORPT	I											
ICORSE	I											
ICPRT	I											
ICPRTY	I											
ICRSN	I											
ICRSPY	I											
ICTME	I											
ICTYPE	I											
IGRAD	I											
IDIDR	I											
IDONER	I											

CROSS REFERENCE SUMMARY C \*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTDB	WRUB	WRUDB	ADCTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
IBLOCN	I												
IBTYPE	I												
IBUCKT	I												
IC	I												
ICBSZE	I												
ICBSZI	I												
ICGRAD	I				C								C
ICING	I												F C
ICLASS	I												
ICLSAD	I												
ICLSID	I												
ICLSTM	I												
ICLSZE	I												
ICOMID	I												
ICONSU	I												
ICORE	I				FSC								
ICORPT	I												
ICORSE	I									C			C
ICPRT	I								C				
ICPTY	I												
ICRSN	I												C
ICRSPY	I												
ICTME	I												
ICTYPE	I								C				
IDGRAD	I												
IDIDR	I				A F								
IDONER	I												

## USAGE SUMMARY

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

SYMBOL	TYPE	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUS	NEWCLS	PBLOCK
IBLOCN	I												
IBTYPE	I						C						
IBUCKT	I				C						C		
IC	I						FC						
ICBSZE	I					C						FC	
ICBSZI	I					C						FC	
ICGRAD	I			C					CE	FC		C	
ICING	I												
ICLASS	I					A FSC						FSC	
ICLSAD	I						C						
ICLSID	I						SC						
ICLSTM	I						FC						
ICLSZE	I						C						
ICOMID	I						SC						
ICONSU	I												
ICORE	I												
ICORPT	I	FC											
ICORSE	I			C			C			C		C	
ICPRT	I						C						
ICPRTY	I		C	C						C		FC	
ICRSN	I												
ICRSPY	I						C						
ICTNE	I												
ICTYPE	I			C			C			C		C	
IDGRAD	I						C						
IDIDR	I		A S				C						
IDONER	I					C	C						



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTEDMP	PUTCLS	PUTPTB	REMCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
IBLOCN	I								A C			
IBTYPE	I								C			C
IBUCKT	I							F C				
IC	I								C			
ICBSZE	I	C		C		C					C	
ICBSZ1	I											C
ICGRAD	I	F C		C		F C						C
ICING	I											
ICLASS	I	F C		SC		FSC						A SC
ICLSAD	I										C	
ICLSID	I											
ICLSTM	I											
ICLSZE	I								A F C		A F C	F C
ICOMID	I	F C										
ICONSU	I											
ICORE	I								A C			
ICORPT	I											
ICORSE	I		F C									
ICORSE	I	C										C
ICPRT	I	A FSC								C		
ICPRTV	I											
ICRSN	I											
ICRSPY	I											
ICTME	I	A FSC									C	
ICTYPE	I											
IDGRAD	I											
IDIDR	I											
IDONER	I	FSC										



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
IBLOCN	I	
IBTYPE	I	
IBUOKT	I	C
IC	I	
ICBSZE	I	FSC
ICBSZ1	I	FSC
ICGRAD	I	C
ICING	I	SC
ICLASS	I	SC
ICLSAD	I	
ICLSID	I	
ICLSTM	I	
ICLSZE	I	
ICOMID	I	
ICONSU	I	
ICORE	I	SC
ICORPT	I	SC
ICORSE	I	C
ICPRT	I	SC
ICPTY	I	C
ICRSN	I	
ICRSPY	I	
ICTME	I	SC
ICTYPE	I	C
IDGRAD	I	
IDIDR	I	
IDONER	I	C

# CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

## USAGE SUMMARY

SYMBOL TYPE

SYMBOL	TYPE	MAIN	SRCTP	SRITDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMNE	UPDATE	WPTB
IDRS	I	F C					C	C		C			
IDSTSK	I							SC					
IDSYNB	I						F C	C					
IDTEGR	I	C					C						
IDTYPE	I												
IDUMP	I	F C					F C	F C		A F C			
IDURAT	I						C	F C					
IEXTRA	I	C					C	C		C			
IFAIL	I	C					SC	C		C			
IFAIL1	I	C					C	C		C			
IFAIL2	I	C					C	C		C			
IFLOW	I	C					F C	F C		F C			
IFRSTC	I	F C					C						
IGID	I	F C					C	C					
IGINTR	I	C					C			C			
ILASTC	I	C					C						
ILPTB	I												
INCORE	I				C							FSC	
INDX1	I				F C		F C					C	
INDX2	I				F C		F C					C	
INTBN	I												
INVRES	I				F C		F C					A C	
IOPTCG	I												
IOPTF	I	C					C	C		C			
IOPTF1	I	C					C	C		C			
IOPTF2	I	C					C	C		C			
IPARM	I							C					

CROSS REFERENCE SUMMARY C \*\*\*\*\* TRAMB \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTDB	WRUB	WRUDB	ADDTQ	ALLOC	ALLOCA	ALLOGD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
IDRS	I				C						C		C
IDSTSK	I												
IDSYNB	I												
IDTEGR	I												
IDTYPE	I				A F								
IDUMP	I				F C					F C			C
IDURAT	I												
IEXTRA	I				C					C			C
IFAIL	I				C					C			C
IFAIL1	I				C					C			C
IFAIL2	I				C					C			C
IFLOW	I				F C					F C		F C	
IFRSTC	I												
IGID	I												
IGINTR	I				C				F C	C			C
ILASTC	I												
ILPTB	I												
INCORE	I												
INDX1	I												
INDX2	I												
INTBN	I												
INVRES	I												
IOPTCG	I												C
IOPTF	I				C					C			C
IOPTF1	I				C					C			C
IOPTF2	I				C					C			C
IPARM	I												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

SYMBOL	TYPE	CLSDMP	CORR	DETLAG	DTRNSF	ERROR	EXECT	FORMC	FORMQ	FREIDE	FRMPTB	GENTDB	GETCLS
IDRS	I		C	C	C		C	F C	FSC				
IDSTSK	I						A F C						
IDSYNB	I		F C		C		C						
IDTEGR	I		C		C		F C				C		
IDTYPE	I								A FSC				
IDUMP	I		F C	C	F C		F C	F C				A F	
IDURAT	I		C		F C		C						
IEXTRA	I		SC	C	C		FSC	C	C				
IFAIL	I		SC	C	C		FSC	C	C				
IFAIL1	I		C	C	C		C	C	C				
IFAIL2	I		C	C	C		C	C	C				
IFLOW	I		F C	C	F C		F C	F C					
IFRSTC	I	F C	C										C
IGID	I		C				C						
IGINTR	I				C			F C	F C	C			
ILASTC	I	F C	C										C
ILPTB	I										C		
INCORE	I			C									
INDX1	I			C									
INDX2	I			C									
INTBN	I												
INVRES	I			A FSC									
IOPTCG	I												
IOPTF	I		C	C	C		F C	C	C				
IOPTF1	I		C	C	C		C	C	C				
IOPTF2	I		C	C	C		F C	C	C				
IPARM	I						C						

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE USAGE SUMMARY

SYMBOL	TYPE	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
IDRS	I			C			C						
IDSTSK	I						FSC						
IDSYNB	I						C						
IDTEGR	I						C						
IDTYPE	I		A S										
IDUMP	I			F C			F C		A F	F C		F C	
IDURAT	I						F C						
IEXTRA	I			C			C			C		C	
IFAIL	I			C			C			C		C	
IFAIL1	I			C			C			C		C	
IFAIL2	I			C			C			C		C	
IFLOW	I			C			F C		F C			F C	
IFRSTC	I					C						FSC	
IGID	I					C	C						
IGINTR	I			C								C	
ILASTC	I					C			F C			FSC	
ILPTB	I	C											
INCORE	I				SC						C		
INDX1	I				C						C		
INDX2	I				C						C		
INTBN	I												
INVRES	I				C								
IOPTCG	I								C		FSC		
IOPTF	I			C								C	
IOPTF1	I			C						C		C	
IOPTF2	I			C						C		C	
IPARM	I						C						



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	PLIST	PREPC	PTBOMP	PUTCLS	PUTPTB	RENCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
IDRS	I		C				C		C		C		C
IDSTSK	I	FC											C
IDSYNB	I											C	AFC
IDTEGR	I												
IDTYPE	I												
IDUMP	I		FC				C		FC	FC	FC		FC
IDURAT	I												FC
IEXTRA	I		C				C		C	C	C		C
IFAIL	I		C				C		SC		C		C
IFAIL1	I		C				C		C	C	C		C
IFAIL2	I		C				C		C	C	C		C
IFLDM	I		FC				C		FC	FC	FC		FC
IFRSTC	I		FC				FSC						C
IGID	I		FSC										
IGINTR	I												
ILASTC	I		C				SC						AFC
ILPTB	I			FC				SC					
INCORE	I					FSC			FSC	C			
INDX1	I								FSC	A	C		
INDX2	I								FSC	A	C		
INTBN	I									A	C		
INVRES	I								AFC	A	FS	T	
IOPTCG	I												
IOPTF	I		C				C			FC	C		C
IOPTF1	I		C				C			C	C		C
IOPTF2	I		C				C			C	C		C
IPARM	I									C			



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
IDRS	I	SC
IDSTSK	I	
IDSYNB	I	
IDTEGR	I	
IDTYPE	I	
IDUMP	I	SC
IDURAT	I	
IEXTRA	I	C
IFAIL	I	C
IFAIL1	I	C
IFAIL2	I	C
IFLOW	I	SC
IFRSTC	I	SC
IGTD	I	C
IGINTR	I	C
ILASTC	I	SC
ILPTB	I	SC
INCORE	I	C
INDX1	I	C
INDX2	I	C
INTBN	I	
INVRES	I	C
IOPTCG	I	SC
IOPTF	I	FSC
IOPTF1	I	FSC
IOPTF2	I	FSC
IPARM	I	

# CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	SRCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRHTE	UPDATE	WPTB
IPBLOC	I											
IPGRAD	I											
IPREDT	I											
IPROCB	I											
IPROP	I											
IPRTY	I											
IPRTYC	I											
IPTYPE	I											
IRBNC	I											
IRSN0	I											
IRUGF	I											
IRUTF	I											
ISAVE	I											
ISORCN	I											
ISORNN	I											
ISORT	I											
ISRCPB	I											
ISRRUB	I											
ISRUDB	I											
ISTASK	I											
ISTATS	I											
ISYNCT	I											
ISZEPT	I											
ISZ1PT	I											
ITASK	I											
ITDATE	I											
ITDBST	I											

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTDB	WRUB	WRUDB	ADDTDQ	ALLOCC	ALLOCA	ALLOCD	ASCLS	ASCLSS	LALQ	CBLOCK	CLASCG
IPBLOC	I												
IPGRAD	I				C								C
IPREDT	I												
IPROCB	I								A F	A F			
IPROP	I												
IPRTY	I												
IPRTYC	I												
IPTYPE	I				C								C
IRBRNC	I												
IRESNO	I												
IRUGF	I												
IRUTF	I												
ISAVE	I												
ISORCN	I												
ISORNN	I					F C		F C					
ISORT	I								C	C			
ISRCPB	I						C						
ISRRUB	I						C						
ISRUDB	I						C						
ISTASK	I						C						
ISTATS	I												
ISYNCT	I												
ISZEPT	I												
ISZIPT	I												
ITASK	I												
ITDATE	I												
ITDAST	I												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRANS \*\*\*\*\* USAGE SUMMARY \*\*\*\*\*

SYMBOL	TYPE	CLSDMP	CORR	DETLAG	DTNNSF	ERRCP	EXFCT	FORMC	FORMQ	PRETD6	PRMP1B	GENTD6	GETCLS
IPBLOC	I				A F C							A F T	
IPGRAD	I				C			FSC	FSC	C			
IPREDT	I		C		C		C				SC		
IPROCB	I		A F C		A C		F C						
IPROP	I												
IPRTY	I		C		C		C						
IPRTYC	I		C		SC		C				C		
IPTYPE	I				C			C A C		C			
IRBRNC	I		C		C		C						
IRESNO	I												
IRUGF	I												
IRUTF	I												
ISAVE	I			FSC									
ISORCN	I		C		C		C				C		
ISORNN	I						C					C	
ISORT	I		C				C						
ISRCPB	I						C					A F C	
ISRRUB	I						C					C	
ISRUDB	I						C					C	
ISTASK	I						C					C	
ISTATS	I		C		C		F C				C		
ISYNCT	I		F C		C		C						
ISZEPT	I										C		
ISZ1PT	I										C		
ITASK	I		C		C		C						
ITDATE	I							A FSC					
ITDBST	I										F C	FSC	

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAN3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY												
		GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	MEMCLS	PBLOCK	
IPBLOC	I								A F ETI					
IPGRAD	I		C	C						F C		C		
IPREDT	I						C							
IPROCB	I						A C							
IPROP	I													
IPRTY	I						C							
IPRTYC	I						C							
IPTYPE	I		C	C						C		C		
IRBRNC	I						C							
IRESDU	I													
IRUGF	I													
IRUTF	I													
ISAVE	I				C					SC				
ISORCN	I						C							
ISORNN	I							FSC						
ISORT	I					C	C							
ISRCPB	I							FSC						
ISRRUB	I							FSC						
ISRUDB	I							FSC						
ISTASK	I						FS							
ISTATS	I						C							
ISYNCT	I						C							
ISZEPT	I	C												
ISZIPT	I	C												
ITASK	I						A F C							
ITDATE	I													
ITDBST	I												F C	

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUTCLS	PUTPTB	REMCLS	REMPTB	KESINV	RESUSE	RUSER	SCATSA	SPLIT
IPBLOC	I											
IPGRAD	I											
IPREDT	I										C	AFC
IPROCB	I											C
IPROP	I				A F							AFC
IPRTY	I											C
IPRTYC	I									C		FC
IPTYPE	I											C
IRBRNC	I											
IREANO	I							A F	AFC			
IRUGF	I								A C			
IRUTF	I								A C			
ISAVE	I							C	SC			
ISORCN	I										C	FC
ISORNN	I									C		
ISORT	I	A FSC										
ISRCPB	I									C		
ISRRUB	I									C		
ISRUDB	I									C		
ISTASK	I									C		
ISTATS	I									C		FSC
ISYNCT	I									C		C
ISZEPT	I		FC		C							
ISZ1PT	I		FC		FC							
ITASK	I											C
ITDATE	I											
ITDBST	I										C	



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
IPBLOC	I	
IPGRAD	I	C
IPREDT	I	
IPROC8	I	
IPROP	I	
IPRTY	I	
IPRTYC	I	
IPTYPE	I	C
IRBRNC	I	
IRSN0	I	
IRUGF	I	
IRUTF	I	
ISAVE	I	C
ISORCN	I	
ISORN	I	
ISORT	I	C
ISRCP8	I	
ISRRUB	I	
ISRUDB	I	
ISTASK	I	
ISTATS	I	
ISYNCT	I	
ISZEPT	I	FSC
ISZIPT	I	FSC
ITASK	I	
ITDATE	I	
ITDBST	I	

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	SRTCTP	SRTTDB	SVRUSI	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMNT	UPDATE	WPTB
ITDURT	I								F C			
ITEVNT	I						C		C			
ITGRDI	I						C		C			
ITIMEC	I						C		C			
ITIMEE	I						C		C			
ITIMES	I						C		C			
ITIME1	I						C					
ITIME2	I						C					
ITRANW	I						C		C			
ITRK1	I								SC			
ITRNRU	I						C		C			
ITSKFN	I											
ITSKTP	I											
ITTYPE	I						F C					
ITI	I	A F	T								C	
IT2	I	A F	T								C	
INTASK	I											
IIFREE	I											
IIPTB	I											
JALTR	I											
JARUDB	I											
JBLKT	I											
JBLOCN	I											
JBTYPE	I											
JCONSU	I											
JDATE	I											
JDTYPE	I											

CROSS REFERENCE SUMMARY \*\*\*\*\* TRAN3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTDB	WRUB	WRUDB	ADDTQ	ALLUC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
ITDURT	I	F C					F C	F C					C
ITEVNT	I				C						C		C
ITGRD1	I				C								C
ITIMEC	I				C						C		C
ITIMEE	I				C						C		C
ITIMES	I				C						C		C
ITIME1	I					A F							
ITIME2	I					A F							
ITRANW	I				C						C		C
ITRKL	I						C	C					
ITRNRU	I				C						C		C
ITSKFN	I												
ITSKTP	I												
ITTYPE	I					A F							
IT1	I												
IT2	I												
IWTASK	I												
IIFREE	I												
IIPTB	I												
JALTR	I												
JARUDB	I												
JBLKT	I												
JBLDCN	I												
JBTYPE	I												
JCONSU	I												
JDATE	I												
JDTYPE	I												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	CLSDMP	CORR	DETLG	DIRNSF	ERRGR	EXECT	FORMC	FORMG	FRETDB	HRMPTB	GENTDB	GETCLS
ITDURT	I											
ITEVNT	I											
ITGRD1	I											
ITIMEC	I											
ITIMEE	I											
ITIMES	I											
ITIME1	I											
ITIME2	I											
ITRANW	I											
ITRK1	I											
ITRNRU	I											
ITSKFN	I											
ITSKTP	I											
ITTYPE	I											
IT1	I											
IT2	I											
INTASK	I											
IIFREE	I											
IIPTB	I											
JALTR	I											
JARUDB	I											
JBLKT	I											
JBLOCN	I											
JBTYPE	I											
JCONSU	I											
JDATE	I											
JDTYPE	I											

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
ITDURT							C					
ITEVNT			C			C			C		C	
ITGRD1		C	C						C		C	
ITIMEC			C			C			C		C	
ITIMEE			C			C			C		C	
ITIMES			C			C			C		C	
ITIME1						C						
ITIME2						C						
ITRANW						C			C		C	
ITRK1			C				C					
ITRNRU			C						C		C	
ITSKFN						C						
ITSKTP						F C						
ITTYPE						C						
IT1		A S		C						C		
IT2				C						C		
IWTASK												
IIFREE								A F T				
I1PTB	C											
JALTR												
JARUDB												
JBLKT												
JBLOCN												
JBTYPE												
JCONSU												
JDATE			C									
JDTYPE			C									

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAMB \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTBUMP	PUTCLS	PUTPTB	REMCLS	REMPTR	RESINV	RESUSE	RUSER	SCATSA	SPLIT
ITDURT	I											
ITEVNT	I	C				C		C	C	C	C	C
ITGRD1	I											
ITIMEC	I	FSC				C		C	F C	C		C
ITIMEE	I	C				C		C	C	C		C
ITIMES	I	C				C		C	C	C		C
ITIME1	I							A F	A F C			
ITIME2	I							A F	A F C			
ITRANW	I	C				C		C	C	C		C
ITRK1	I											
ITRNRU	I	C				C		C	C	C	F C	C
ITSKFN	I											
ITSKTP	I								C	C		
ITTYPE	I								C			
IT1	I							FSC	C			
IT2	I							FSC	C			
IWTASK	I											
ILFREE	I											
ILPTB	I		F C		SC		FSC					
JALTR	I								F C			
JARUDB	I							A SC				
JBLKT	I								C			
JBLOCN	I							A	C			
JBTYPE	I								C			
JCONSU	I							A	C			
JDATE	I											
JDTYPE	I											



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
ITDURT	I	
ITEVNT	I	C
ITGRD1	I	C
ITIMEC	I	SC
ITIMEE	I	A FSC
ITIMES	I	FSC
ITIME1	I	
ITIME2	I	
ITRANW	I	FSC
ITRK1	I	
ITRNRU	I	FSC
ITSKFN	I	
ITSKTP	I	
ITTYPE	I	
IT1	I	C
IT2	I	C
IWTASK	I	
IIFREE	I	SC
I1PT8	I	SC
JALTR	I	
JARUDB	I	
JBLKT	I	
JBLOCN	I	
JBTYP	I	
JCONSU	I	
JDATE	I	C
JDTYPE	I	C

CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

SYMBOL	TYPE	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMNT	UPDATE	WPTB
JID	I												
JNTBN	I												
JRESNO	I												
JRUGF	I												
JRUTF	I												
JTTYPE	I												
KEOF	I	C					C	C					
KLASZE	I							SC					
KTIME1	I							FSC					
KTIME2	I							FSC					
LAGC	I	F C					C	C					
LAGT	I	C					C						
LAGTOT	I	C					C						
LBRNCH	I							C					
LJMS	I				F C							C	
LJMITC	I	C					C						
LJMIT1	I												
LJMS	I				F C							C	
LJMNXT	I				C							C	
LJMPTB	I												
LJMR	I				C							C	
LJINTR	I												
LJTIM1	I							C					
LJTIM2	I							C					
LOTIM1	I				C							A C	
LOTIM2	I				C							A C	
LRSON	I	C					C						

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTDB	WRUB	WRUDB	ADDTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLUCK	CLASCG
JID	I												
JNTBN	I												
JRESNO	I												
JRUGF	I												
JRUTF	I												
JTTYPE	I												
KEOF	I				C								C
KLASZE	I												
KTIME1	I												
KTIME2	I												
LAGC	I								C				
LAGT	I												
LAGTOT	I												
LBRNCH	I												
LIMIS	I												
LIMITC	I												
LIMIT1	I												
LIMNS	I												
LIMNXT	I												
LIMPTB	I												
LIMRES	I												
LIMTR	I												
LITIM1	I												
LITIM2	I												
LOTIM1	I												
LOTIM2	I												
LRSON	I												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	CLSDMP	CORR	DETLAG	DTRNSF	ERROR	EXEC	FORMC	FORMQ	FRETDB	FRMPTB	GENTDB	GETCLS
JID							AFC					
JNTBN												
JRESNO												
JRUGF												
JRUTF												
JTTYPE							AFC					
KEOF												
KLASZE												
KTIME1												
KTIME2												
LAGC												
LAGT												
LAGTOT												
LBRNCH												
LIMIS												
LIMITC												
LIMIT1												
LIMNS												
LIMNXT												
LIMPTB												
LIMRES												
LIMTR												
LITIM1												
LITIM2												
LOTIM1												
LOTIM2												
LRSON												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
JID	I											
JNTBN	I											
JRESNO	I											
JRUGF	I											
JRUTF	I											
JTTYPE	I											
KEOF	I											
KLASZE	I											
KTIME1	I											
KTIME2	I											
LAGC	I											
LAGT	I											
LAGTOT	I											
LBRNCH	I											
LIMIS	I											
LIMITC	I											
LIMIT1	I											
LIMNS	I											
LIMNXT	I											
LIMPTB	I											
LIMRES	I											
LIMTR	I											
LITIM1	I											
LITIM2	I											
LOTIMI	I											
LOTIM2	I											
LRSON	I											



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

## USAGE SUMMARY

[illegible]



CROSS REFERENCE SUMMARY C \*\*\*\*\* TRANS \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
JID	I	C
JNTBN	I	
JRESNO	I	
JRUGF	I	
JRUTF	I	
JTTYPE	I	C
KEOF	I	C
KLASZE	I	
KTIME1	I	
KTIME2	I	
LAGC	I	C
LAGT	I	
LAGTOT	I	
LBRNCH	I	
LIMIS	I	SC
LIMITC	I	FSC
LIMIT1	I	FSC
LIMNS	I	SC
LIMNXT	I	SC
LIMPTB	I	FSC
LIMRES	I	SC
LIMTR	I	SC
LITIM1	I	
LITIM2	I	
LOTIM1	I	C
LOTIM2	I	C
LRSON	I	

CROSS REFERENCE SUMMARY C \*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMTE	UPDATE	WPTB
LTIME1	I						FSC					
LTIME2	I						FSC					
MAXCLS	C					F C			C			
MAXLAG	C					C			C			
MINGRO	F C					C			C			
MRUDBS	I											
MXTRUB	I											
NACLS	F C					C						
NBI	I			C	C						C	
NBUCKT	I			C	C						C	
NCGRPS	I											
NCING	I											
NCLSES	I					C			C			
NCOURS	I					C			C			
NCRSES	I					C			F C			
NDXCLS	I					F C						
NDXND1	I					C			C			
NDXTDL	I					C			C			
NDXTD1	I					C			C			
NEXTPT	I											
NLBRNC	I					C						
NLFTB	I								F C			
NOPB	I					C						
NOSRCS	I						F C					
NOSTDS	I					F C			FSC			
NOTDRS	I					C						
NPARMS	I								C			

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRANS \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		WRLTOB	WRUB	WRUDB	ADDTDO	ALLDC	ALLOCA	ALLOGD	ASCLS	ASCLSS	CALQ	CBLOCK	CLASCG
LTIME1	I												
LTIME2	I												
MAXCLS	I				C								F C
MAXLAG	I				C					C			C
MINGRD	I				C					C			C
MRUDBS	I												
MXTRUB	I												
NACLS	I							F C	F C				
NBI	I									A F			
NBUCKT	I												
NCCRPS	I												F C
NCING	I												F C
NCLSES	I				C					C			C
NCOURS	I				F C								C
NCRSES	I				C					C			C
NDXCLS	I												
NDXNDI	I				C								C
NDXTDL	I				FSC								C
NDXTDI	I				FSC								C
NEXTPT	I												
NLBRNC	I												
NLFTB	I	F C					C						
NOPB	I							F C	F C				
NOSRCS	I						C						
NOSTDS	I												
NOTDRS	I											C	C
NFARMS	I												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	CLSDMP	CORR	DETLAG	DTRNSF	ERROR	EXEC	FORMC	FORMQ	FRETD6	FRMPTB	GENTDB	GETCLS
LTIME1						F C						
LTIME2						F C						
MAXCLS				C			C	C	C			
MAXLAG		F C	C	C		C	C	C				
MINGRD		C	C	C		C	FSC	A FSC				
MRUDBS												
MXTRUB												
NACLS		C				C						
NBI			C									
NBUCKT			C									
NCGRPS												
NCING												
NCLSES		C	C	C		C	C	C	C			
NCOURS				C		C	C	C				
NCRSES		C	C	C		C	F C	F C				
NDXCLS		F C		A C		A FSC				C		
NDXND1				C			C	C	C			
NDXTDL				C			C	C	3C			
NDXTD1				C			C	C	FSC			
NEXTPT												
NLBRNC		C		F C		C				F C	FSC	
NLFTB												
NOPB		C				C						
NOSRCS						C						
NDSTDS		F C		C		F C					F C	
NOTDRS		C	C	C		C		FSC				
NPARMS						C						

CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	GETPTB	GETTDB	GRADE	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
LTIME1	I											
LTIME2	I											
MAXCLS	I	C	C			SC			FC		C	
MAXLAG	I		C			SC					C	
MINGRD	I		C			C					C	
MRUDBS	I					C						
MXTRUB	I											
NACLS	I				C							
NBI	I			C								
NBUCKT	I			C								
NCGRPS	I			C								
NCING	I											
NCLSES	I											
NCOURS	I											
NCRSES	I											
NDXCCLS	I											
NDXNDI	I											
NDXTDL	I											
NDXTDI	I											
NEXTPT	I											
NLBRNC	I											
NLFTB	I											
NOPB	I											
NGSRCS	I											
NOSTDS	I											
NOTDRS	I											
NPARMS	I											

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	PLIST	PREPC	PTBDMP	PUTCLS	PUTPTB	RECHLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
LTIME1	I	F C											
LTIME2	I	F C											
MAXCLS	I						C		C	C			C
MAXLAG	I						C		C	C			C
MINGRD	I								F C				
MRUDBS	I												
MXTRUB	I								C				
NACLS	I		A FSC						FSC	A C			
NBI	I								SC	C			
NBUCKT	I												
NCGRPS	I												
NCING	I												
NCLSES	I		F C				FSC		C	C			C
NCOURS	I												
NCRSES	I		C				C		C	C			C
NDXCLS	I										A F C	A C	
NDXND1	I												
NDXTDL	I												
NDXTD1	I												F C
NEXTPT	I												F C
NLRNC	I												
NLFTB	I											C	
NOPB	I		FSC										
NDSRCS	I												
NOSTDS	I										A F C	F C	
NOTORS	I		C				C		C	C			C
NPARMS	I												



GROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
LTIME1	I	
LTIME2	I	
MAXCLS	I	C
MAXLAG	I	FSC
MINGRD	I	SC
MRUDBS	I	
MXTRUB	I	
NACLS	I	SC
NBI	I	C
NBUCKT	I	C
NCGRPS	I	FSC
NCING	I	SC
NCLSES	I	SC
NCIDURS	I	F C
NCRSES	I	SC
NDXCLS	I	
NDXND1	I	SC
NDXTDL	I	SC
NDXTD1	I	SC
NEXTPT	I	
NLBRNC	I	
NLFTB	I	
NDPB	I	C
NOSRCS	I	
NOSTDS	I	
NOTDRS	I	SC
NPARMS	I	

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAN3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	MAIN	SRTCTP	SRTTDB	SVRUS1	SVRUS2	SYNC	SYNCT	TBLOCK	TRACKD	TRMNT	UPDATE	WPTB
NPROC	I	C				A F C						
NRBRNC	I					C	C					
NRESCR	I			C							F C	
NRUDBS	I											
NSAVE	I			F C	FSC						C	
NSRCE	I								F C			
NSYNCT	I						SC					
NTASKS	I					C	F C					
NTDBRL	I								A FSC			
NTSKS	I						FSC					
NUMBLK	I								F C			
NUMCRS	I	A				C						
NUMCRU	I											
NUMGRD	I					C						
NUMSTA	I								C			
NUMTR	I											
NXT	I											
NXTBRK	I	FSC		C		F C	C		C		SC	A F T
NXTFPT	I											
NXTFRE	I					C						
NXTNDA	I								F C			
NXTRUB	I											
PB	CB					C	C					
PROP	R											
PTB	CB											
PTBC	CB											
RES	CB			C	C						C	

CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY	WRLTDB	WRUB	WRUDB	ADTDQ	ALLOD	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ	CBLGCK	CLASCG
NPROCB	I													
NRBRNC	I													
NRESCR	I													
NRUDBS	I													
NSAVE	I													
NSRCE	I		F C					F C	F C					
NSYNCT	I													
NTASKS	I													
NTDBRL	I		C					C	C					
NTSKS	I													
NUMBLK	I		F C					C	C					
NUMCRS	I													
NUMCRU	I													
NUMGRD	I													
NUMSTA	I		C					FSC	FSC					A FSC
NUMTR	I													
NXT	I													
NXTBRK	I													
NXTFPT	I											C		C
NXTFRE	I													
NXTNDA	I													
NXTRUB	I		F C					C	C					
PB	CB													
PROP	R													
PTB	CB													
PTBC	CB													
RES	CB													

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	CLSDMP	CORR	DETLAG	DTRNSF	ERROR	EXEC	FORMC	FORMQ	FRETDB	FRMPYB	GENTDB	GETCLS
NPROCB	I	A F C		A SC		A C				F C		
NRBRNC	I	C		C		C						
NRESCR	I		C									
NRUDBS	I											
NSAVE	I		C									
NSRCE	I											
NSYNCT	I					C				C	FSC	
NTASKS	I					C						
NTDBRL	I											
NTSKS	I									C	FSC	
NUMBLK	I					F C						
NUMCRS	I									F C	FSC	
NUMCRU	I					A F C				C		
NUMGRD	I											
NUMSTA	I					C				C		
NUMTR	I									F C	C	
NXT	I						A FSC					
NXTBRK	I		A F C									
NXTFPT	I		C			C		C				
NXTFRE	I									C		
NXTNDA	I											C
NXTRUB	I									F C	FSC	
PB	CB											
PROP	R					C						
PTB	CB											
PTBC	CB											
RES	CB		C							C		

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
NPROCB	I						A F C	A F					
NRBRNC	I						C						
NRESCR	I				C					C			
NRUDBS	I												
NSAVE	I				C					FSC			
NSRCE	I							C					
NSYNCT	I						SC						
NTASKS	I						F C						
NTDBRL	I							C					
NTSKS	I						FSC		FS				
NUMBLK	I							F C					
NUMCRS	I						A C						
NUMCRU	I								C				
NUMGRD	I						C						
NUMSTA	I							C					
NUMTR	I												
NXT	I				SC					C			
NXTBRK	I						C			C		C	
NXTEPT	I												
NXTFRE	I	C									FSC		
NXTNDA	I					C							
NXTRUB	I							C					
PB	CB												
PROP	R												
PTB	CB												
PTBC	CB	C											
RES	CB				C							C	



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUTCLS	PUTPTB	RENCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
NPROCB	I										A C	A F C
NRBRNC	I											C
NRESCR	I							C	C			
NRUDBS	I							F C				
NSAVE	I							C	SC			
NSRCE	I										C	
NSYNCT	I											
NTASKS	I											C
NTDBRL	I										C	
NTSKS	I											
NUMBLK	I										C	
NUMCRS	I										A F C	A F C
NUMCRU	I											
NUMGRD	I										C	F C
NUMSTA	I										SC	
NUMTR	I											
NXT	I											
NXTBRK	I											C
NXTFPT	I											
NXTFRE	I											C
NXTNDA	I											
NXTRUB	I										C	
PB	CB											
PROP	R											C
PTB	CB											F C
PTBC	CB											C
RES	CB											



CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
NPROCB	I	
NRBRNC	I	
NRESCR	I	C
NRUDBS	I	
NSAVE	I	C
NSRCE	I	
NSYNCT	I	
NTASKS	I	
NTDBRL	I	
NTSKS	I	
NUMBLK	I	
NUMCRS	I	
NUMCRU	I	SC
NUMGRD	I	
NUMSTA	I	
NUMTR	I	C
NXT	I	C
NXTBRK	I	SC
NXTFPT	I	SC
NXTFRE	I	SC
NXTNDA	I	
NXTRUB	I	
PB	CB	
PROP	R	
PTB	CB	
PTBC	CB	C
RES	CB	C

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAN3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	MAIN	SRTCTP	SRTTDB	SVRUSI	SVRUSZ	SYNC	SYNCT	TBLOCK	TRACKD	TRMTE	UPDATE	WPTB
RLTDBC									C			
RUB												
RUDB												
SORDSC									C			
STUDNO												
STUDSN												
TB							C					
TDR												
TLIST							C					
WORKB												
WRKA												

GROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY									
		WRLTDB	WRUB	WRUDB	ADDTDQ	ALLOC	ALLOCA	ALLOCD	ASCLS	ASCLSS	CALQ
RLTDBC	CB	C					C	C			
RUB	CB										
RUBB	CB										
SORDSC	CB						C	C			
STUDNO	R										
STUDSN	R										
TB	CB										
TDR	CB										
TLIST	CB										
WORKB	CB										
WRKA	CB										

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	USAGE SUMMARY											
		CLSDMP	CORR	DETLG	DTRNSF	ERROR	EXECT	FORMC	FORMQ	FRETDB	FRMPTB	GENTDB	GETCLS
RLTDBC	CB										C		
RUB	CB												
RUDB	CB												
SORDSC	CB						C		A FSC		C		
STUDNO	R												
STUDSN	R							A F C					
TB	CB						C						
TDR	CB								C				
TLIST	CB						C						
WCRKB	CB				C								
WRKA	CB												

CROSS REFERENCE SUMMARY C\*\*\*\*\* TRAM3 \*\*\*\*\*

SYMBOL	TYPE	GETPTB	GETTDB	GRADF	INITR	LAG	LSTASK	LSTRAK	LSTSRC	MLTCLS	NEGUSE	NEWCLS	PBLOCK
RLTDBC	CB							C					
RUB	CB												
RUOB	CB												
SORDSC	CB								C				
STUDNO	R												
STUDSN	R			F C									
TB	CB												
TDR	CB						C						
TLIST	CB						C						
WORKB	CB												
WRKA	CB			C									

CROSS REFERENCE SUMMARY C\*\*\*\*\*  
SYMBOL TYPE

USAGE SUMMARY

	PLIST	PREPC	PTBDM	PUICLS	PUTPTB	REMCLS	REMPTB	RESINV	RESUSE	RUSER	SCATSA	SPLIT
RLTDBC	CB											
RUB	CB								C			
RJDB	CB								C			
SORDSC	CB											
STUDNO	R											
STUDSN	R											
TB	CB											
TDR	CB								C			
TLIST	CB											
WORKB	CB											
WRKA	CB											



## CROSS REFERENCE SUMMARY

SYMBOL TYPE

## USAGE SUMMARY

	RDNAME	*BLOCK	NAME	NUMBER	CLOCK	BLKIN	BLOCK	INTRES	INTSOR	UPDRES	UPDSOR	GETRES
BLKS	CB											
CBLK	CB											
CONTRL	CB											
IAVAIL	I											
IBLOCK	I											
IBUOKT	I											
ICTYPE	I											
IGU	I											
IEGRAD	I											
IFIRST	I											
IGRAD	I											
ILAST	I											
IPEROD	I											
IPRIOR	I											
IPTYPE	I											
IQUANT	I											
ITIME	I											
ITIMEH	I											
ITIMEL	I											
ITYPE	I											
ITL	I											
IUNIT	I											
IMORD	I											
JUNIT	I											
LBLOCK	I											
LINK	I											
MAVAIL	I											

## CROSS REFERENCE SUMMARY

## USAGE SUMMARY

SYMBOL	TYPE	GETSOR	PUTRES	PUTSOR
BLKS	CB			
CBLK	CB			
CONTRL	CB			
IAVAIL	I	C	FSC	FSC
IBLOCK	I			
IBUCKT	I	F C	F C	F C
ICTYPE	I			
ICU	I	C		C
IEGRAD	I			
IFIRST	I	F C	FSC	FSC
IGRAD	I			
ILAST	I	C	FSC	FSC
IPEROD	I			
IPRIOR	I			
IPTYPE	I			
IQUANT	I	F C	SC	SC
ITIME	I	F C	FSC	FSC
ITIMEH	I	F C	F C	F C
ITINEL	I	F C	F C	F C
ITYPE	I			
ITI	I		A F	A F
IUNIT	I			
IWORD	I			
JUNIT	I			
LBLOCK	I			
LINK	I	F C	FSC	FSC
MAVAIL	I	C	C	C

CROSS REFERENCE SUMMARY \*\*\*\*\* TRAM3 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	INIT
RLTDBC	CB	
RUB	CB	
RUDB	CB	
SORDSC	CB	
STUDNO	R	
STUDSN	R	C
TB	CB	
TDR	CB	
TLIST	CB	
WORKB	CB	
WRKA	CB	C

## CROSS REFERENCE SUMMARY

SYMBOL TYPE

## USAGE SUMMARY

		RDNAME	#BLOCK	NAME	NUMBER	CLOCK	BLKIN	BLOCK	INTRES	INTSOR	UPDRES	UPDSOR	GETRES
MAXNUM	I	SC	D C	C	C								
MXSIZE	I						SC	C					
NAM	CB	C	C	C	C								
NAME	I				A F T								
NAMES	I	SC	C	F C	F C								
NAVAIL	I												
NBLOCK	I								FSC	FSC	FSC	FSC	C
NCOURS	I						SC	C					
NRES	I						SC	C					
NSOR	I								FSC		F C	F C	C
NTYPE	I	SC	D C	F C	F C					FSC			
NUM	I	SC	C	F C	F C								
NUMBER	I			A F									
RES	CB								C		C		C
RSOURC	CB								C		C		C
SOR	CB											C	
SOURCE	CB									C		C	

# CROSS REFERENCE SUMMARY

## USAGE SUMMARY

SYMBOL	TYPE	GETSUR	PUTRES	PUTSOR
MAXNUM	I			
MXSIZE	I			
NAM	CB			
NAME	I			
NAMES	I			
NAVAIL	I	C	FSC	FSC
NBLOCK	I			
NCOURS	I			
NRES	I		C	
NSOR	I	C		C
NTYPE	I			
NUM	I			
NUMBER	I			
RES	CB		C	
RSOURC	CB		C	
SOR	CB	C		C
SOURCE	CB	C		C

Section 4.0  
PHASE 4 PROGRAMMER'S GUIDE

Section 4.1  
INTRODUCTION

The purpose of Phase 4 is to report the resource usage of the training system and to compute the associated costs.

This manual is intended to aid the programmer in the operation and modification of the computer program. It is assumed that the reader of this manual is already familiar with the contents of Technical Memorandum SAT-5, TRAM User's Guide.



## Section 4.2

### PROGRAM DESCRIPTION

The first processing performed by phase 4 is to read the card inputs and print them. The primary resources defined by the card inputs are then matched with the primary resources passed from phase 2 via file 24. The bucket sizes from that file complete the primary resource specifications from the card inputs. There must be a one to one correspondence between the resources from phase 2 and for those for phase 4. The secondary resources defined for phase 4 are completely independent of the other TRAM job steps.

The program then starts reading the use records from unit 40. The data from these records are stored in two separate common areas. One is for the periodic report, and the other is for the yearly report. Also, a plot bucket record is written to unit 51 for each primary resource that is to be plotted. The contents of these records will be discussed later. The program continues reading and processing the use records until the time for the next report, or the end of the run is reached.

The periodic report is produced at the specified frequency. This report consists of a printout of the information stored in the periodic report common variables. After the report is printed, the common area is cleared out for the next report. Note that this report is completely independent of the yearly report, and can be produced at any specified frequency.

The yearly report not only summarizes the resource usage, but also includes the costs associated with that usage. These costs are computed at the end of each year and stored for the final cost summary. A separate yearly report is printed for primary and secondary resources. At this time, RGU plot data are stored for those resources that are to be plotted. In addition, a plot bucket record is written for secondary resources (the bucket size for all secondary resources is one year.)

When the end of the run is reached, a final periodic and yearly report are printed, even if these reports would not normally be due at this time. The final cost summary is then printed. This report shows the costs that were incurred in each category for each year. They are shown in both current dollar

values and in inflated values. If RDT&E costs have been incurred in years prior to the start of the run, they will be shown in year zero and negative years.

The final processing that is done is to produce the use plots. The data for these plots have been stored throughout the run. RGU data, which consists of time, number of RGUs on hand, and actual use available, have been stored in common /RGU/. The two temporary files contain the rest of the required information. These are the plot bucket files that were referred to earlier. Their records contain the time, use, and maximum use available for the resources. The data contained on these files are retrieved and combined with the RGU data to produce the plots. Note that for secondary resources, the maximum use available is the same as the actual use available, since there is no maximum use restriction on secondary resources.

### Section 4.3

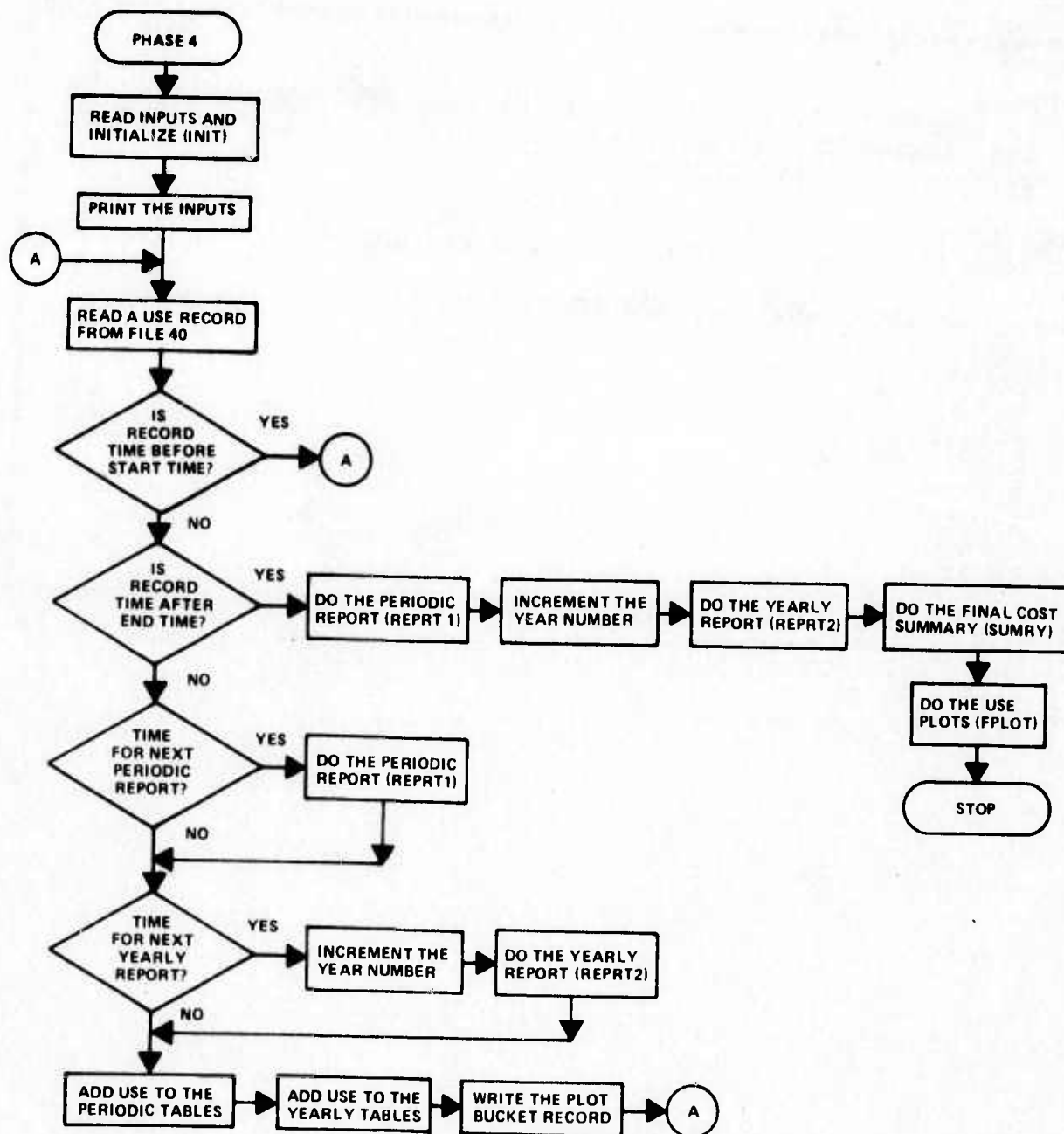
#### SUBPROGRAM DESCRIPTIONS

This section contains the descriptions of the individual subprograms that comprise phase 4 of the TRAM program. The description for each subprogram consists of a statement of the purpose of the routine, the calling sequence, a description of its parameters, the method used, and a list of the subprograms required. A high level flowchart, which shows the logical decision points and the processing accomplished, is also included for each of the major subprograms.

```

CC***** PHASE 4 *****
CC*
CC*  PURPOSE
CC*    TO REPORT THE TIME HISTORY OF TRAINING RESOURCE USAGE WHICH
CC*    WAS PASSED FROM PHASE 3, AND COMPUTE THE COSTS ASSOCIATED
CC*    WITH THE TRAINING SYSTEM
CC*
CC*  REFERENCES
CC*    TRAM USERS GUIDE AND TRAM PROGRAMMERS GUIDE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    INIT
CC*    PRINT1
CC*    PRINT2
CC*    PRINT3
CC*    REPT1
CC*    REPT2
CC*    WPLOTB
CC*    SUMRY
CC*    FPLOT
CC*****

```



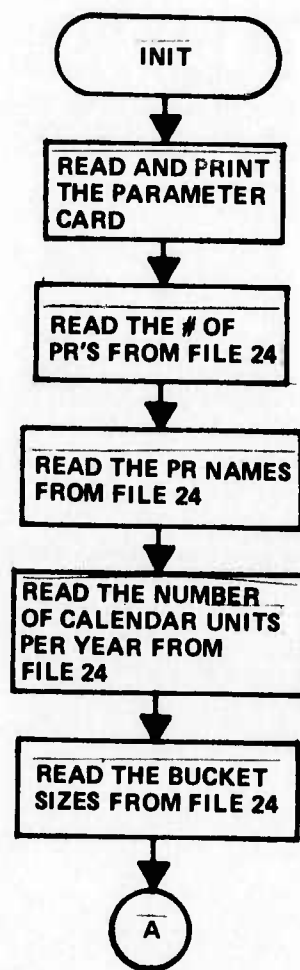
**PHASE 4 MAIN PROGRAM**

```

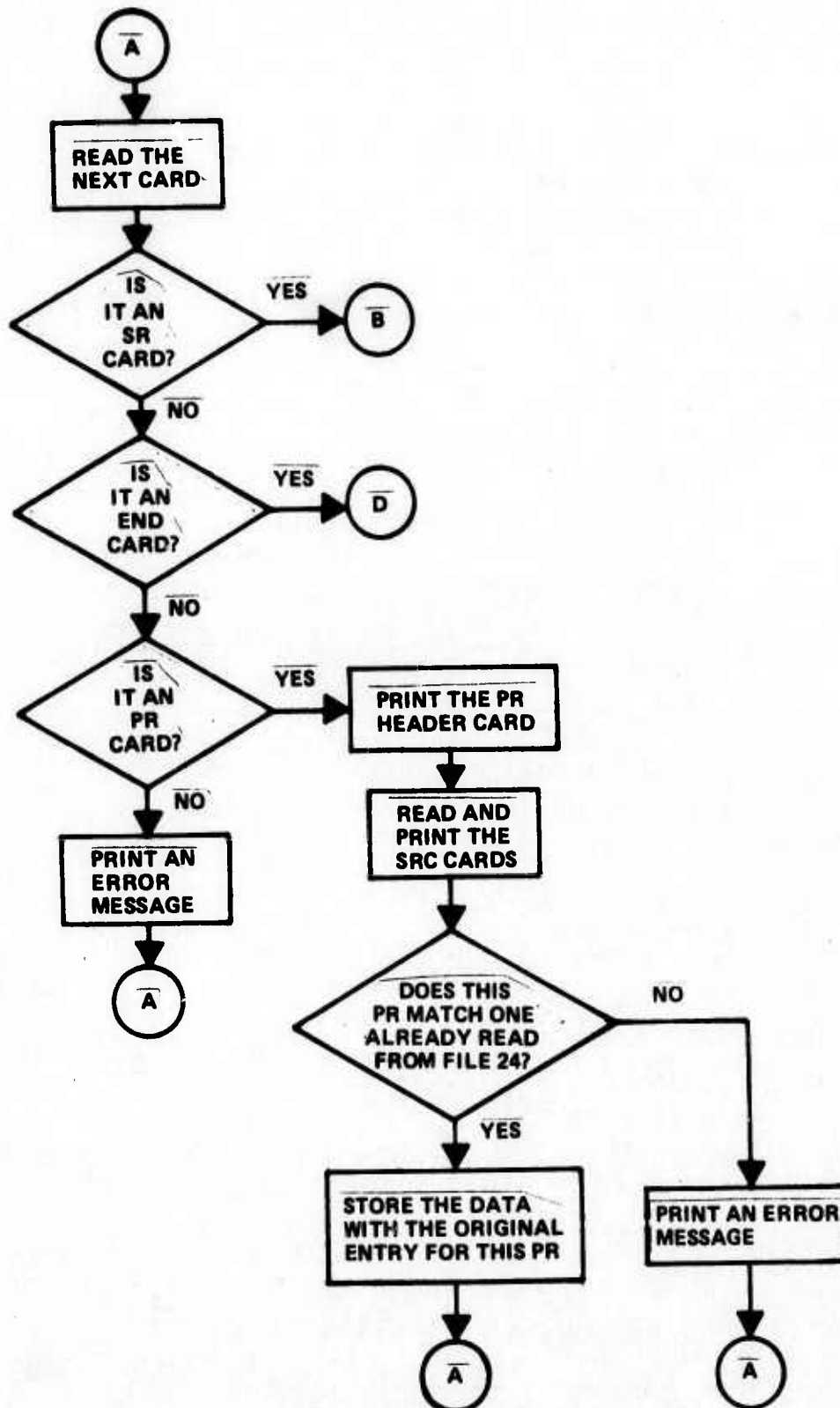
CC***** INIT *****
CC*
CC*          SUBROUTINE INIT
CC*
CC*  PURPOSE
CC*    TO INITIALIZE STEP 4 OF TRAM. THIS INVOLVES THE FOLLOWING:
CC*    1 READ PARAMETERS CARD
CC*    2 READ PRIMARY RESOURCE DEFINITIONS PASSED FROM STEP 3
CC*    3 READ PRIMARY AND SECONDARY RESOURCE DATA FROM CARDS
CC*    4 SET RESOURCE USEAGE COUNTS AND COST SUMMARIES TO ZERO
CC*    5 SET UP TIME OF FIRST PERIODIC AND YEARLY REPORTS
CC*
CC*  CALLING SEQUENCE
CC*    CALL INIT
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    CLEAR
CC*    LOOKUP
CC*    LOOK2
CC*
CC*****

```

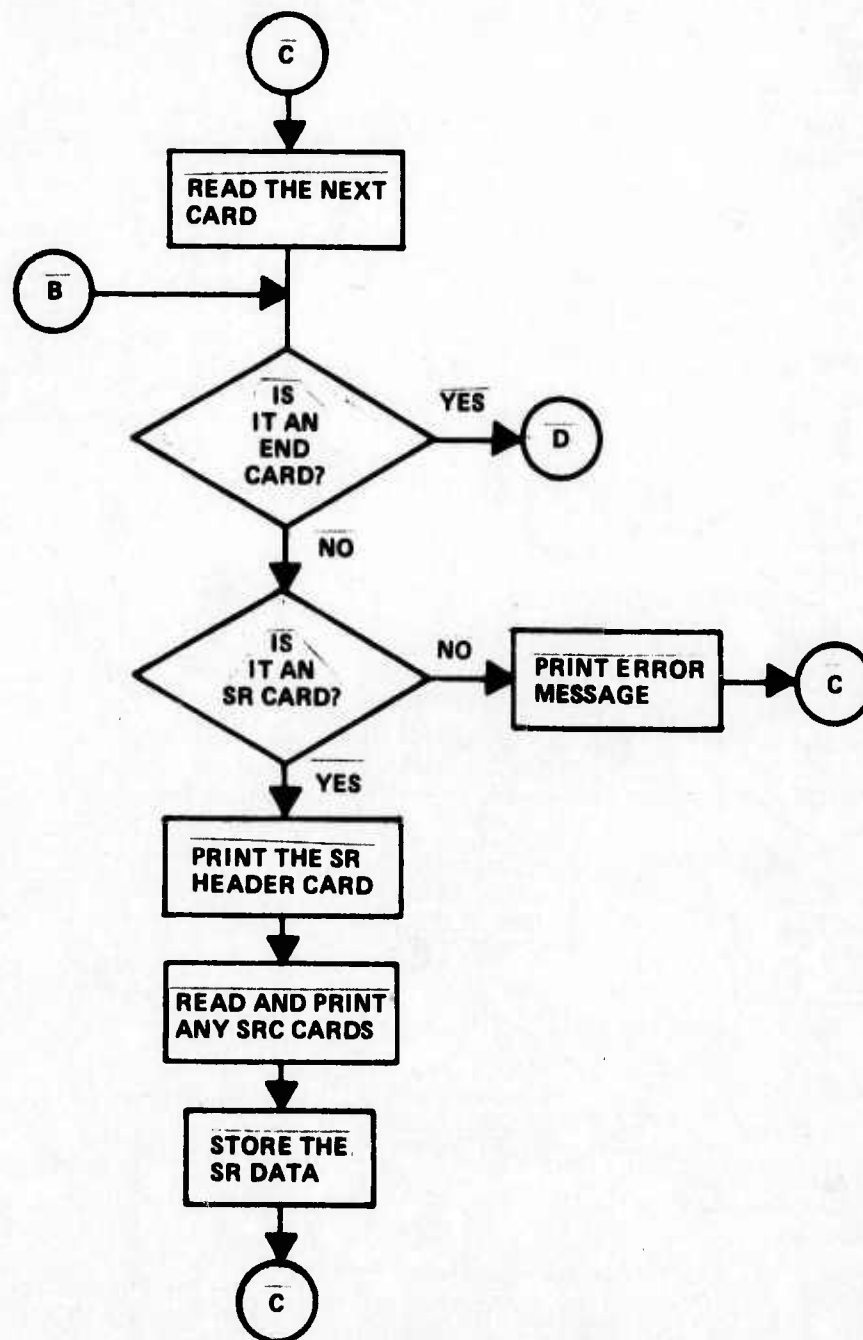




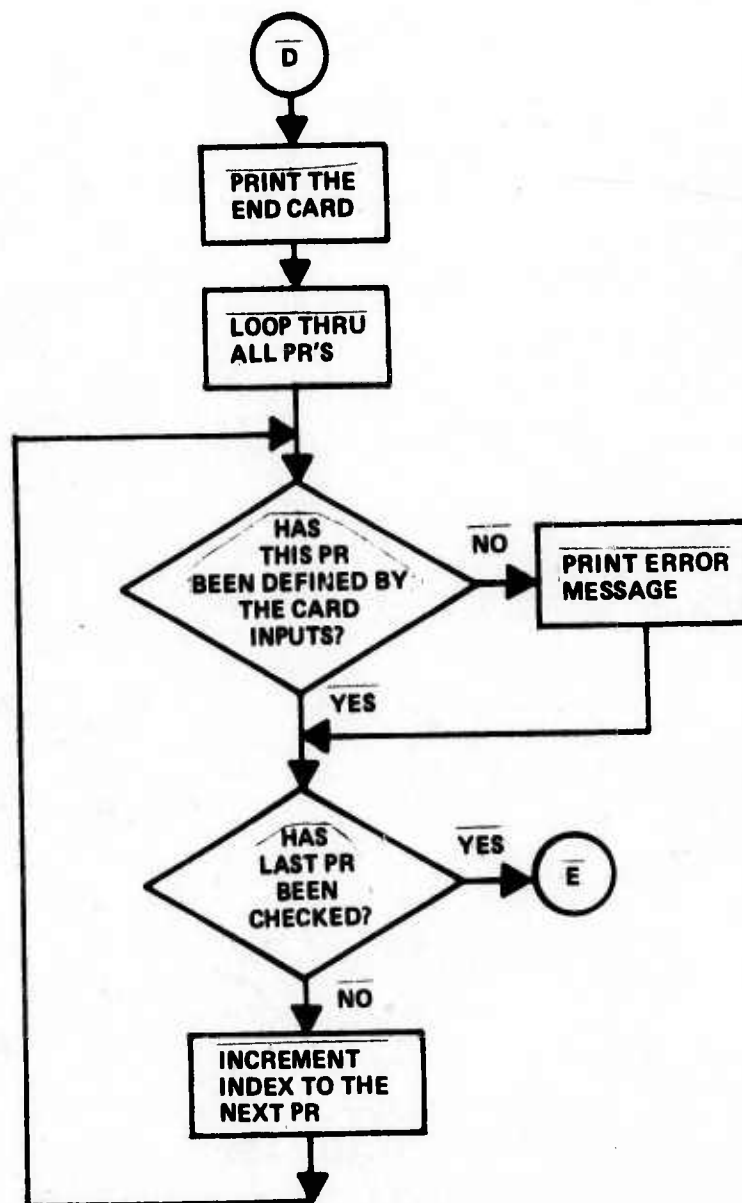
**SUBROUTINE INIT**



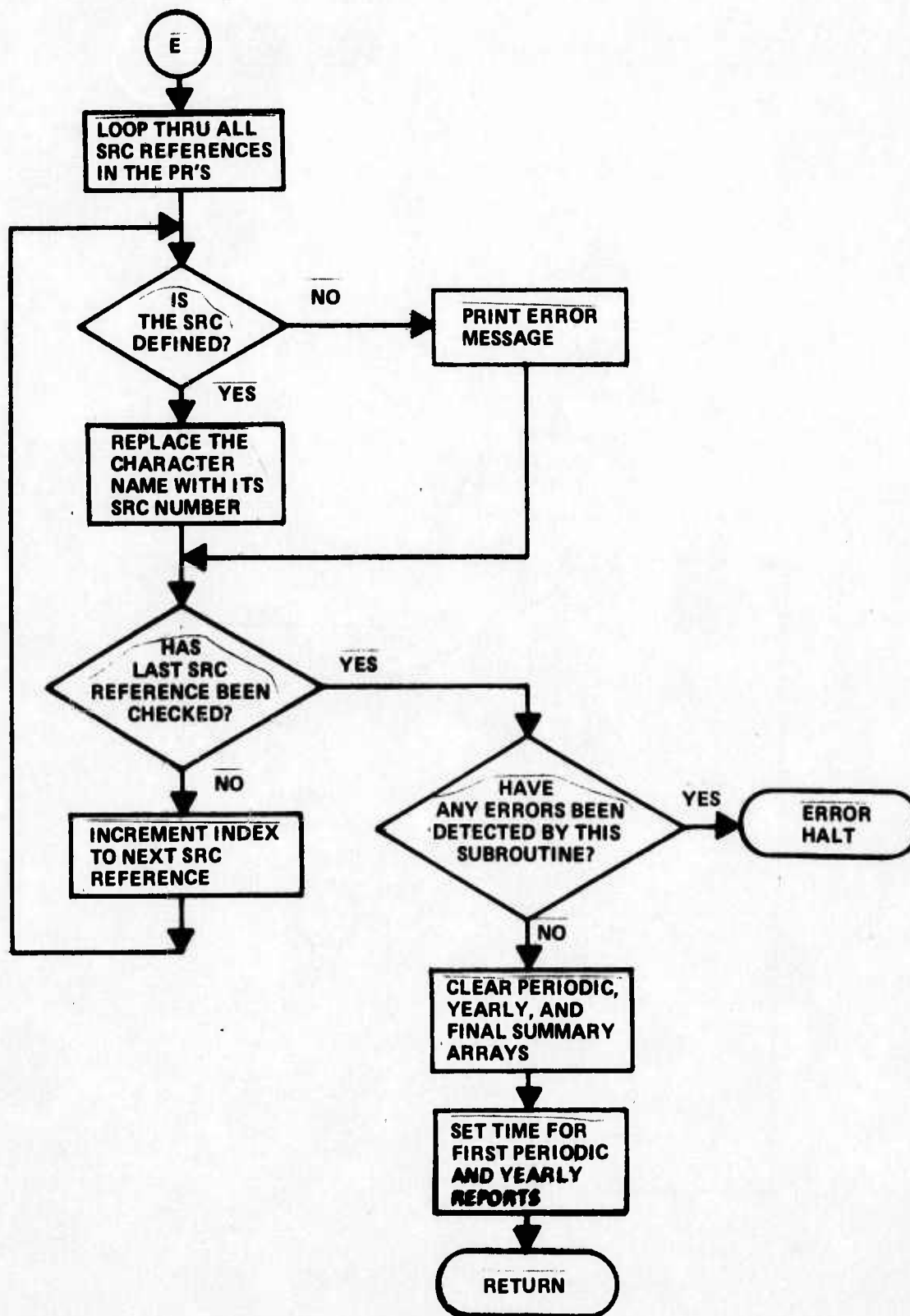
SUBROUTINE INIT -- CONTINUED



SUBROUTINE INIT - CONTINUED



SUBROUTINE INIT - CONTINUED



SUBROUTINE INIT - CONTINUED

```

CC***** CLEAR *****
CC*
CC*          SUBROUTINE CLEAR
CC*
CC*  PURPOSE
CC*    TO CLEAR AN ARRAY TO ZERO
CC*
CC*  CALLING SEQUENCE
CC*    CALL CLEAR (IARRAY, NWDS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    IARRAY - ARRAY TO BE CLEARED
CC*    NWDS   - NUMBER OF ELEMENTS IN IARRAY TO BE CLEARED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```



```

CC***** BLOCKD *****
CC*
CC*          BLOCK DATA
CC*
CC*  PURPOSE
CC*    TO INITIALIZE COMMON AREAS FOR TRAM STEP 4
CC*
CC*****

```

```

CC***** LOOK2 *****
CC*
CC*          SUBROUTINE LOOK2
CC*
CC*  PURPOSE
CC*    TO LOOK UP A VALUE IN A TABLE AND RETURN ITS POSITION. THE
CC*    TABLE CONSISTS OF THOSE ELEMENTS IN A TWO DIMENSIONAL ARRAY
CC*    WHICH HAVE A CERTAIN FIXED FIRST SUBSCRIPT.
CC*
CC*  CALLING SEQUENCE
CC*    CALL LOOK2 (IVAL, IARRAY, N1, N2, N1FIX, ICODE, INDEX)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IVAL  - VALUE TO BE SEARCHED FOR
CC*      IARRAY - TABLE OF VALUES
CC*      N1    - DIMENSION OF FIRST SUBSCRIPT OF IARRAY
CC*      N2    - DIMENSION OF SECOND SUBSCRIPT OF IARRAY
CC*      N1FIX  - FIRST SUBSCRIPT OF VALUES IN IARRAY TO BE SEARCHED
CC*      ICODE  - DATA TYPE
CC*                1  INTEGER
CC*                2  CHARACTER
CC*    OUTPUT
CC*      INDEX - POSITION (SECOND SUBSCRIPT) OF THE VALUE IN THE
CC*                TABLE (ZERO IF NOT FOUND)
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** PRINT1 *****
CC*
CC*          SUBROUTINE PRINT1
CC*
CC*  PURPOSE
CC*    TO PRINT THE INPUT PARAMETERS FOR TRAM STEP 4
CC*
CC*  CALLING SEQUENCE
CC*    CALL  PRINT1
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** PRINT2 *****
CC*
CC*          SUBROUTINE PRINT2
CC*
CC*  PURPOSE
CC*    TO PRINT A TABLE OF THE PRIMARY RESOURCE DEFINITIONS
CC*
CC*  CALLING SEQUENCE
CC*    CALL PRINT2
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** PRINT3 *****
CC*
CC*
CC*          SUBROUTINE PRINT3
CC*
CC*  PURPOSE
CC*    TO PRINT A TABLE OF THE SECONDARY RESOURCE DEFINITIONS
CC*
CC*  CALLING SEQUENCE
CC*    CALL  PRINT3
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*****

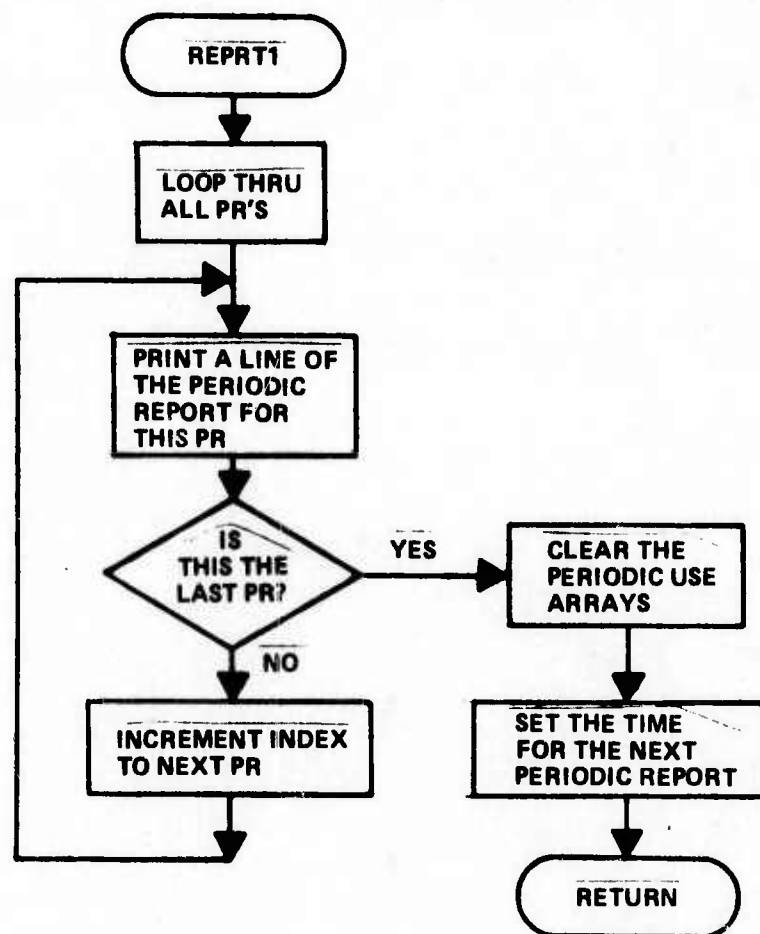
```

```

CC***** REPT1 *****
CC*
CC*          SUBROUTINE REPT1
CC*
CC*  PURPOSE
CC*    TO PRINT THE PERIODIC REPORT
CC*
CC*  CALLING SEQUENCE
CC*    CALL REPT1
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    NONE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    CLEAR
CC*
CC*****

```



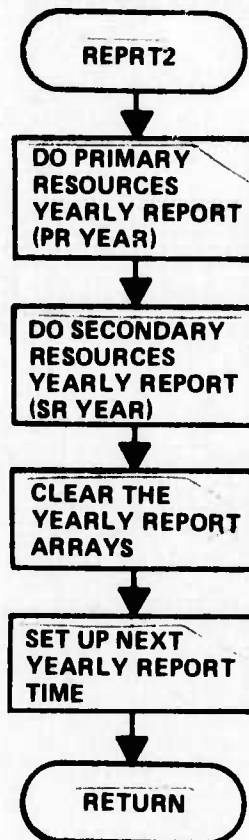


SUBROUTINE REPRT1

```

CC***** REPRT2 *****
CC*
CC*          SUBROUTINE REPRT2
CC*
CC*  PURPOSE
CC*    TO COMPUTE THE COSTS INCURRED BY THE RESOURCE USEAGE DURING
CC*    THE YEAR AND TO PRINT THE YEARLY SUMMARY REPORT
CC*
CC*  CALLING SEQUENCE
CC*    CALL REPRT2 (IYEAR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*    IYEAR  - YEAR NUMBER OF THIS REPORT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    CLEAR
CC*    PRYEAR
CC*    SRYEAR
CC*
CC*****

```

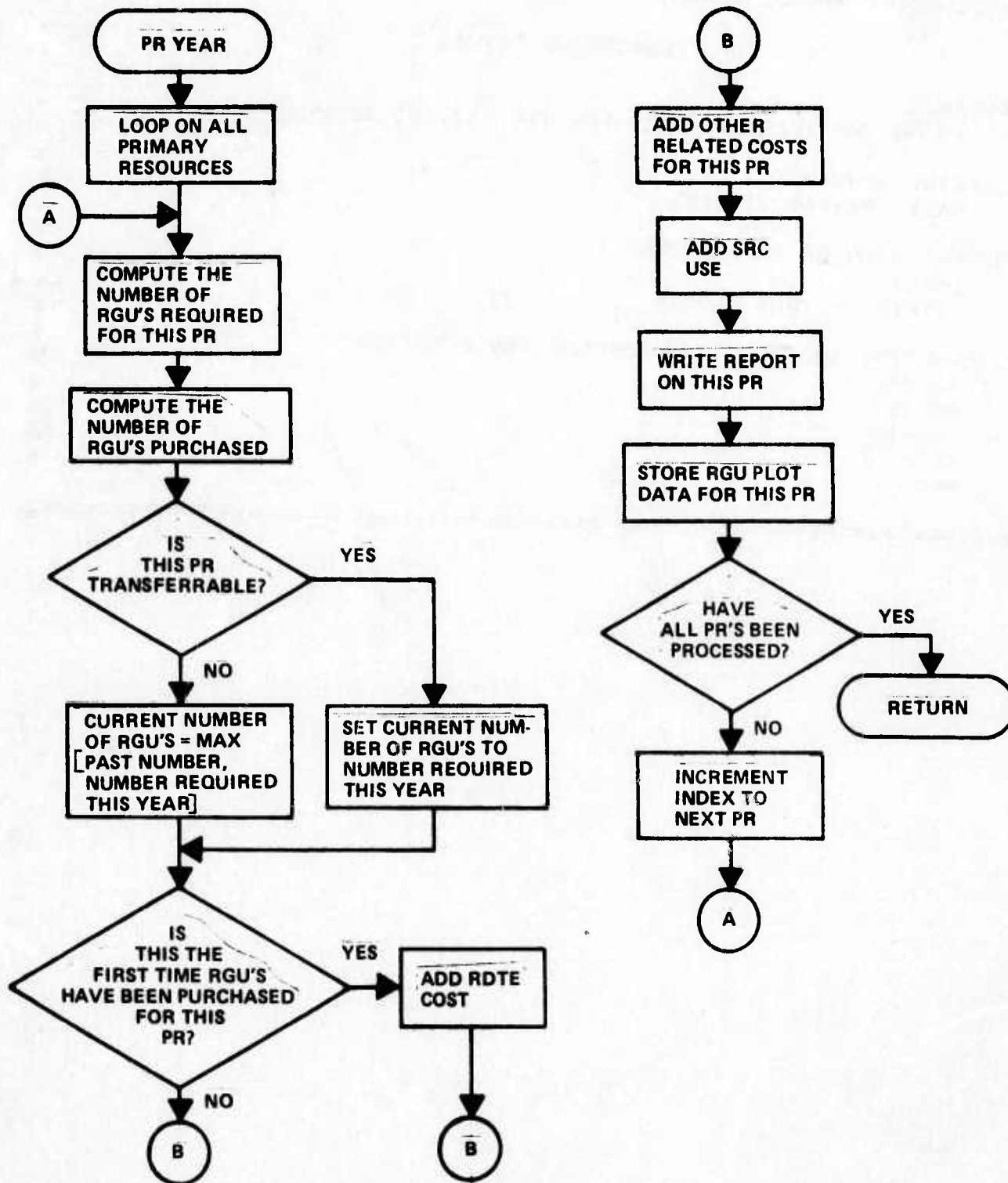


**SUBROUTINE REPRT2**

```

CC***** PRYEAR *****
CC*
CC*          SUBROUTINE PRYEAR
CC*
CC*  PURPOSE
CC*    TO DO THE YEARLY REPORT FOR THE PRIMARY RESOURCES
CC*
CC*  CALLING SEQUENCE
CC*    CALL  PRYEAR (IYEAR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*    IYEAR  - YEAR NUMBER
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    IICOST
CC*    RGCOST
CC*    PRINT4
CC*    RDTE
CC*    WRGU
CC*
CC*****

```



SUBROUTINE PRYEAR

```

CC***** PRINT4 *****
CC*
CC*          SUBROUTINE PRINT4
CC*
CC*  PURPOSE
CC*    TO PRINT THE PRIMARY RESOURCE YEARLY REPORT FOR SUBROUTINE
CC*    PRYEAR
CC*
CC*  CALLING SEQUENCE
CC*    CALL PRINT4 (I,IYEAR,IVAL)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      I      - NUMBER OF THE PRIMARY RESOURCE WHICH THIS CALL IS
CC*              FOR, OR ZERO TO INITIALIZE A NEW REPORT
CC*      IYEAR  - YEAR NUMBER
CC*      IVAL   - ARRAY OF VALUES FOR THE PRIMARY RESOURCE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

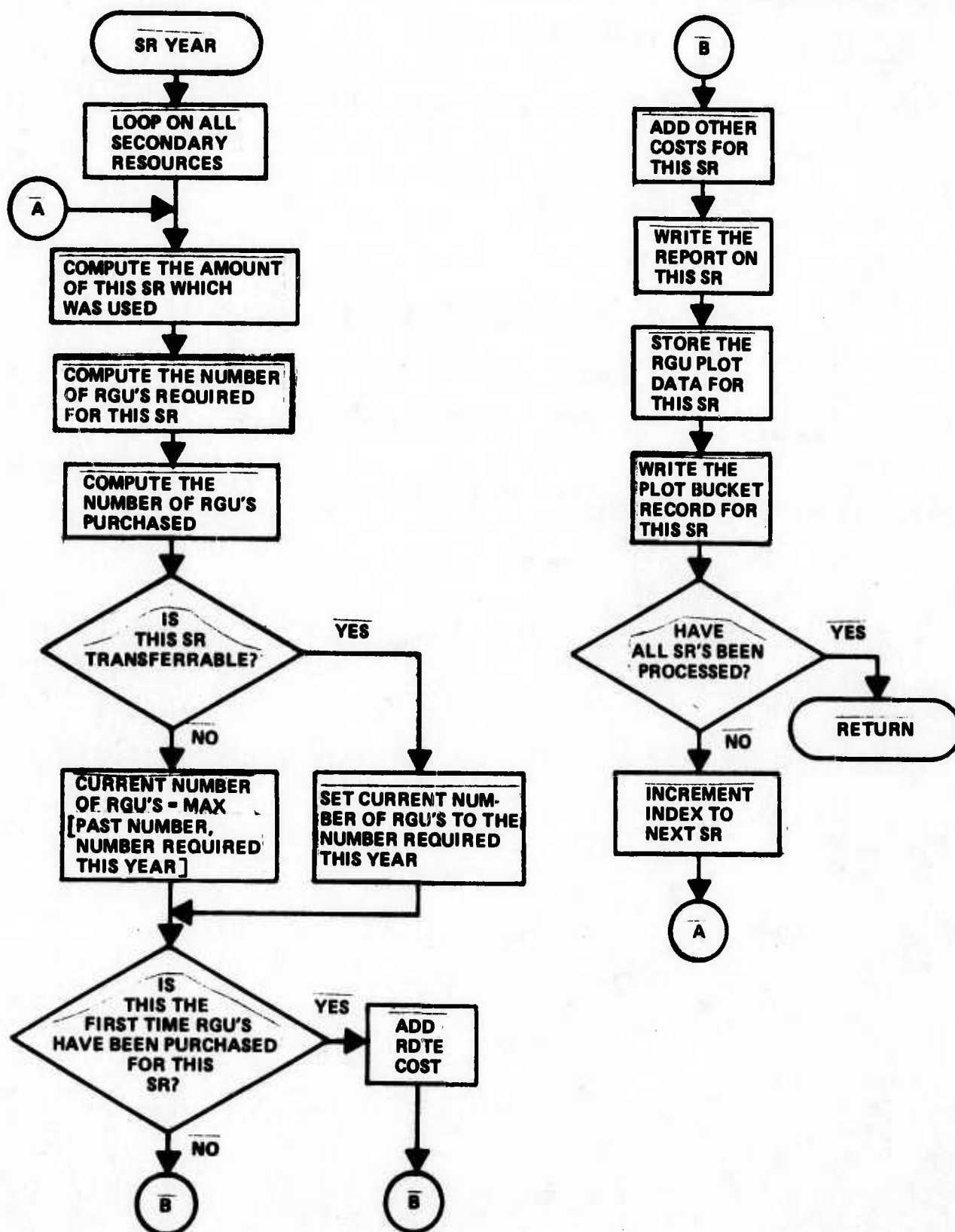
```



```

CC***** SRYEAR *****
CC*
CC* SUBROUTINE SRYEAR *
CC*
CC* PURPOSE *
CC* TO DO THE YEARLY REPORT FOR THE SECONDARY RESOURCES *
CC*
CC* CALLING SEQUENCE *
CC* CALL SRYEAR (IYEAR) *
CC*
CC* DESCRIPTION OF PARAMETERS *
CC* INPUT *
CC* IYEAR - YEAR NUMBER *
CC*
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED *
CC* IICOST *
CC* RGCOST *
CC* PRINT5 *
CC* RTE *
CC* WRGU *
CC* WPLOTB *
CC*
CC*****

```



SUBROUTINE SR YEAR

```

CC***** PRINT5 *****
CC*
CC*          SUBROUTINE PRINT5
CC*
CC*  PURPOSE
CC*    TO PRINT THE SECONDARY RESOURCE YEARLY REPORT FOR SUBROUTINE
CC*    SRYEAR
CC*
CC*  CALLING SEQUENCE
CC*    CALL PRINT5 (I,IYEAR,IVAL)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      I      - NUMBER OF THE SECONDARY RESOURCE WHICH THIS CALL IS
CC*              FOR, OR ZERO TO INITIALIZE A NEW REPORT
CC*      IYEAR  - YEAR NUMBER
CC*      IVAL   - ARRAY OF VALUES FOR THE SECONDARY RESOURCE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** RDTE *****
CC*
CC*          SUBROUTINE RDTE
CC*
CC*  PURPOSE
CC*    TO ADD THE RDTE COST TO THE COST SUMMARY.  RDTE COST IS
CC*    SPREAD OUT OVER THE N YEARS PREECEDING THE CURRENT YEAR
CC*
CC*  CALLING SEQUENCE
CC*    CALL  RDTE (IYEAR, ICOST, N)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IYEAR  - YEAR NUMBER OF THE CURRENT YEAR
CC*      ICOST  - RDTE COST TO BE SPREAD OVER N YEARS
CC*      N      - NUMBER OF YEARS OVER WHICH ICOST IS TO BE INCURRED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** IICOST *****
CC*
CC*          SUBROUTINE IICOST
CC*
CC*  PURPOSE
CC*    TO COMPUTE THE COST ASSOCIATED WITH THE PURCHASE OF RESOURCE
CC*    GENERATOR UNITS
CC*
CC*  CALLING SEQUENCE
CC*    CALL IICOST (II, NRGUP, NRGUH, ICOST)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      II      - INITIAL INVESTMENT COST PER RGU
CC*      NRGUP   - NUMBER OF RGUS PURCHASED
CC*      NRGUH   - NUMBER OF RGUS ALREADY ON HAND
CC*    OUTPUT
CC*      ICOST   - INITIAL INVESTMENT COST
CC*
CC*  METHOD
CC*    SUBROUTINE RGCOST IS CALLED TO COMPUTE THE COST AS FOLLOWS
CC*      1 IF II .LT. 0
CC*        COST = NRGUP * IABS(II)
CC*      2 IF II .GT. 0
CC*        COST = C2 - C1
CC*        C2 = N * II * RL ** LOG2(N)
CC*        C1 = NRGUH * II * RL ** LOG2(NRGUH)
CC*        N = NRGUH + NRGUP
CC*        RL = LEARNING RATE FROM PARAMETER CARD
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    RGCOST
CC*****

```



```

CC***** RGCOST *****
CC*
CC*          SUBROUTINE RGCOST
CC*
CC*  PURPOSE
CC*    TO COMPUTE THE COST ASSOCIATED WITH HAVING RESOURCE
CC*    GENERATOR UNITS ON HAND
CC*
CC*  CALLING SEQUENCE
CC*    CALL RGCOST (II, NRGU, ICOST)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      II      - COST PER RGU (NEGATIVE TO PREVENT LEARNING RATE
CC*                FROM BEING APPLIED)
CC*      NRGU    - NUMBER OF RESOURCE GENERATOR UNITS
CC*    OUTPUT
CC*      ICOST   - COST
CC*
CC*  METHOD
CC*    IF II .LT. 0
CC*      COST = NRGU * IABS(II)
CC*    IF II .GT. 0
CC*      COST = NRGU * II * RL ** LOG2(NRGU)
CC*      RL = LEARNING RATE FROM PARAMETER CARD
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

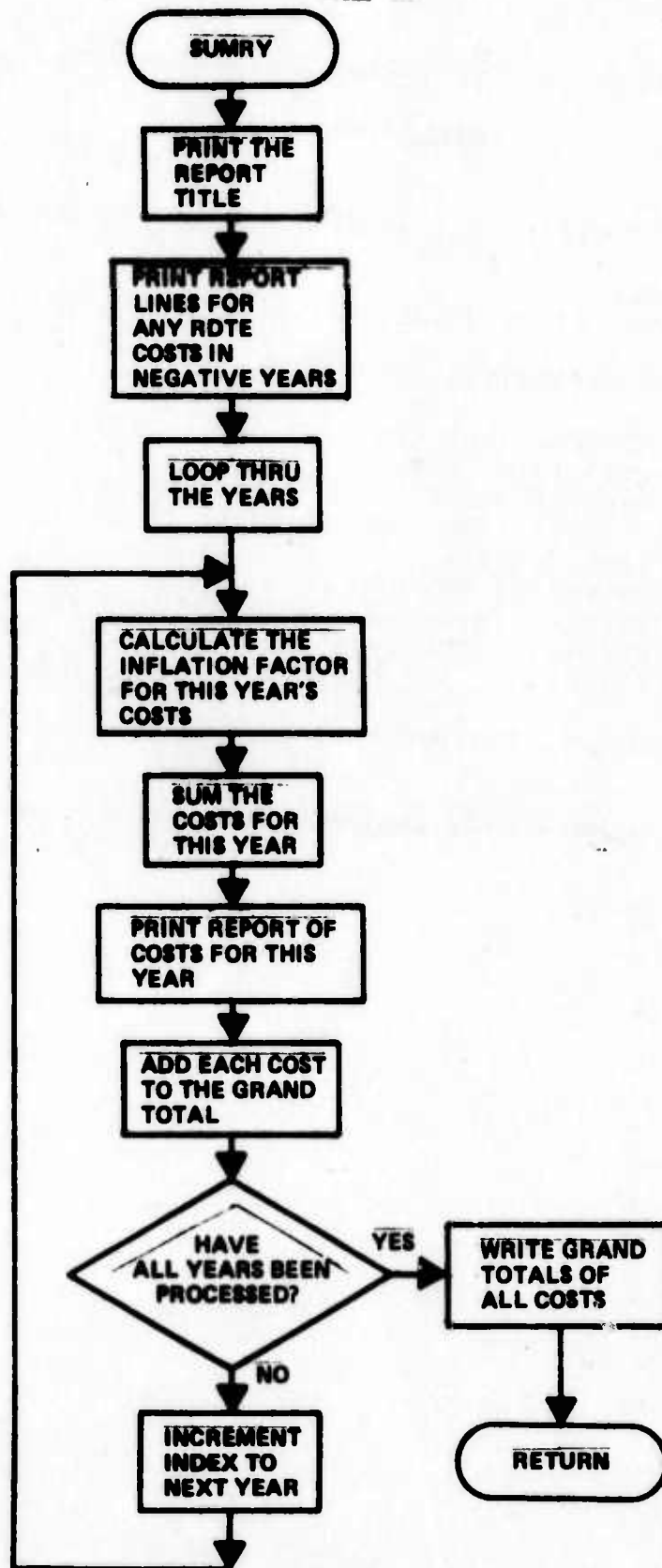
```



```

CC***** SUMRY *****
CC*
CC*          SUBROUTINE SUMRY
CC*
CC*  PURPOSE
CC*    TO PRINT THE FINAL COST SUMMARY REPORT
CC*
CC*  CALLING SEQUENCE
CC*    CALL  SUMRY (NYEARS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*    NYEARS - NUMBER OF YEARS BEING SUMARIZED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PRINT6
CC*****

```



**SUBROUTINE SUMRY**

```

CC***** PRINT6 *****
CC*
CC*          SUBROUTINE PRINT6
CC*
CC*  PURPOSE
CC*    TO PRINT THE FINAL COST SUMMARY REPORT FOR SUBROUTINE SUMRY
CC*
CC*  CALLING SEQUENCE
CC*    CALL PRINT6 (ISW, IYEAR, IRTDE, II, IRI, IOM)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ISW      - CONTROL VARIABLE
CC*                LESS THAN ZERO TO INITIALIZE THE REPORT
CC*                ZERO TO PRINT FINAL TOTALS
CC*                GREATER THAN ZERO TO PRINT SUMMARY FOR A SINGLE YEAR
CC*      IYEAR    - YEAR NUMBER
CC*      IRTDE    - RTDE COST FOR THE YEAR
CC*      II       - INITIAL INVESTMENT COST FOR THE YEAR
CC*      IRI      - RECURING INVESTMENT COST FOR THE YEAR
CC*      IOM      - OPERATIONS AND MAINTENANCE COST FOR THE YEAR
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** WRGU *****
CC*
CC*          SUBROUTINE WRGU
CC*
CC*  PURPOSE
CC*    TU STORE THE RGU PLOT DATA IN COMMON /RGU/
CC*
CC*  CALLING SEQUENCE
CC*    CALL WRGU (IRES,IYEAR,NRGU,IAA)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IRES  -  PRIMARY RESOURCE NUMBER, OR THE NEGATIVE OF THE
CC*               SECONDARY RESOURCE NUMBER
CC*      IYEAR -  YEAR NUMBER
CC*      NRGU  -  NUMBER OF RESOURCE GENERATING UNITS ON HAND
CC*      IAA   -  ACTUAL NUMBER OF USE UNITS AVAILABLE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** WPLOTB *****
CC*
CC*          SUBROUTINE WPLOTB
CC*
CC*  PURPOSE
CC*    TO WRITE OUT A PLOT BUCKET RECORD TO THE PRIMARY RESOURCE
CC*    FILE OR TO THE SECONDARY RESOURCE FILE
CC*
CC*  CALLING SEQUENCE
CC*    CALL WPLOTE (IRES, ITIME, IUSE, IMA)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IRES  -  PRIMARY RESOURCE NUMBER, OR THE NEGATIVE OF THE
CC*               SECONDARY RESOURCE NUMBER
CC*      ITIME -  BUCKET END TIME IN CALENDAR UNITS
CC*      IUSE  -  NUMBER OF UNITS OF THE RESOURCE USED DURING THE
CC*               BUCKET
CC*      IMA   -  MAXIMUM NUMBER OF USE UNITS AVAILABLE FOR THE
CC*               RESOURCE DURING THE BUCKET
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

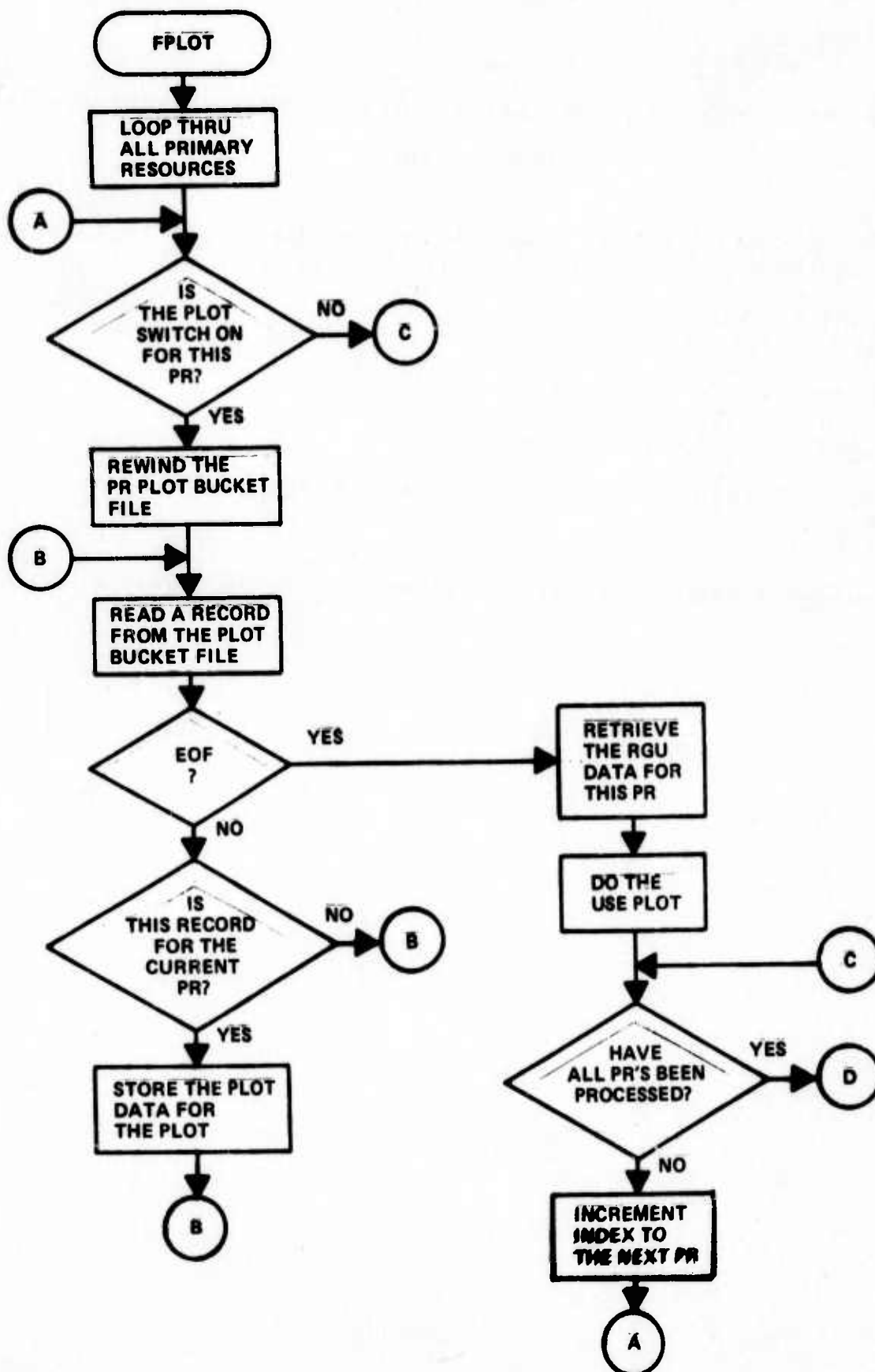
```

```

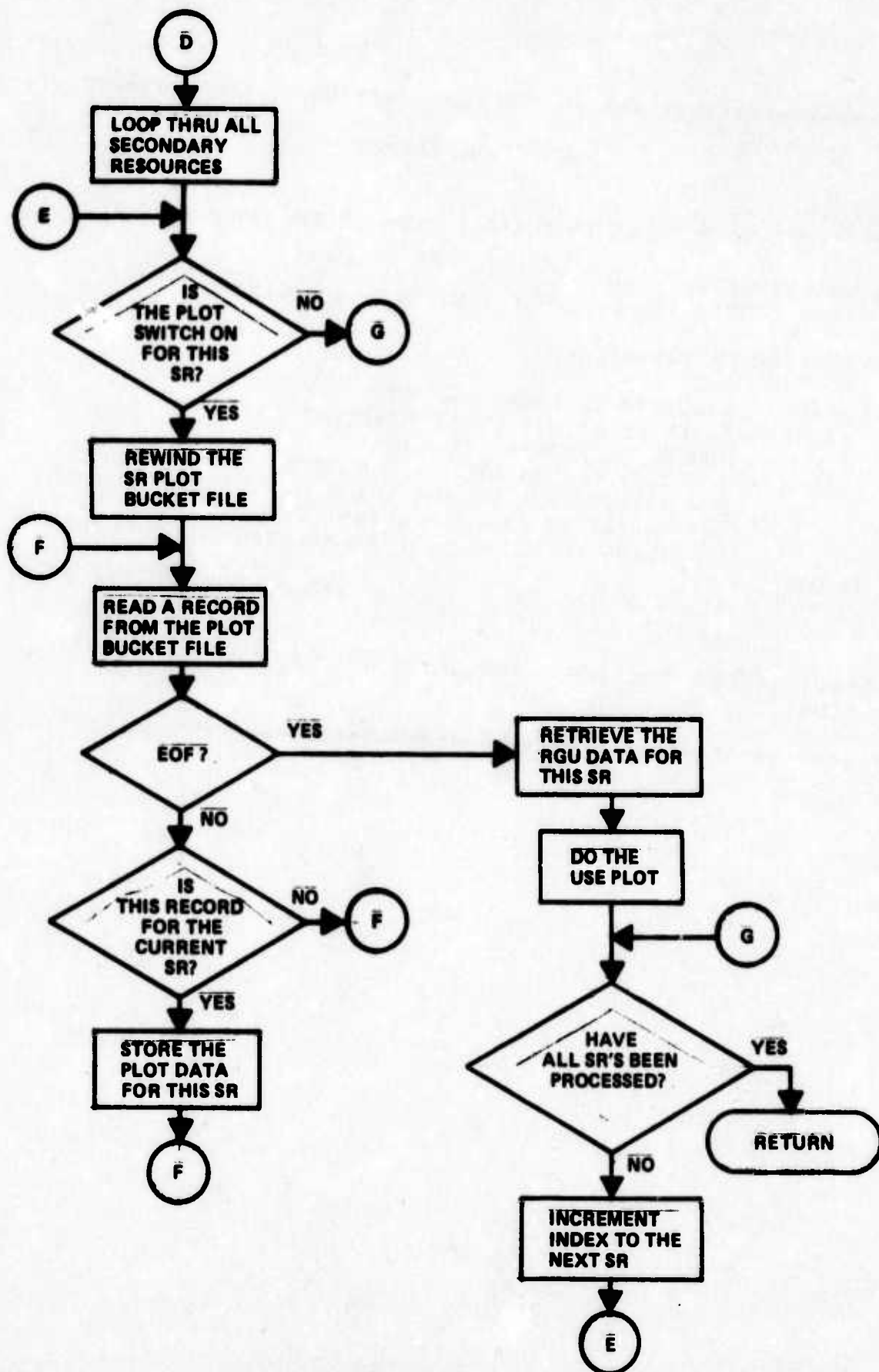
CC***** FPLOT *****
CC*
CC*          SUBROUTINE FPLOT
CC*
CC*  PURPOSE
CC*    TO RETRIEVE THE PLOT DATA FOR EACH RESOURCE THAT IS TO BE
CC*    PLOTTED AND CALL SUBROUTINE PLOTU TO DO THE PLOTS
CC*
CC*  CALLING SEQUENCE
CC*    CALL  FPLOT (IYEAR)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*    IYEAR  -  NUMBER OF YEARS IN THE RUN
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PLOTU
CC*    EFPLOT
CC*
CC*****

```





**SUBROUTINE FPLOTT**



SUBROUTINE FPLOT - CONTINUED

```

CC***** LOOKUP *****
CC*
CC*          SUBROUTINE LOOKUP
CC*
CC*  PURPOSE
CC*    TO LOOK UP A VALUE IN A TABLE AND RETURN ITS POSITION
CC*
CC*  CALLING SEQUENCE
CC*    CALL LOOKUP (IVAL, IARRAY, N, ICODE, INDEX)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      IVAL  - VALUE TO BE SEARCHED FOR
CC*      IARRAY- TABLE OF VALUES TO BE SEARCHED FOR
CC*      N     - NUMBER OF ENTRIES IN IARRAY
CC*      ICODE - 1 - DATA VALUES OCCUPY ONE WORD
CC*              2 - DATA VALUES OCCUPY THREE WORDS, USED FOR 16
CC*                  CHARACTER FIELDS ON IBM COMPUTER
CC*                  (REQUIRES IVAL(3), IARRAY(3,N) )
CC*
CC*    OUTPUT
CC*      INDEX - INDEX OF THE VALUE IN THE TABLE, ZERO IF THE VALUE
CC*              IS NOT FOUND
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*
CC*****

```

```

CC***** PLOTU *****
CC*
CC*          SUBROUTINE PLOTU
CC*
CC*  PURPOSE
CC*    TO PLOT THE RESOURCE USE PLOTS
CC*
CC*  CALLING SEQUENCE
CC*    CALL PLOTU (XMA,USE,TIME1,NPTS1,AA,RGUS,TIME2,NPTS2,NAME)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    XMA      - MAXIMUM USE AVAILABLE PER BUCKET ARRAY
CC*    USE      - USE PER BUCKET ARRAY
CC*    TIME1    - END   TIME OF BUCKET FOR EACH VALUE OF XMA AND USE
CC*    NPTS1    - NUMBER OF ELEMENTS IN XMA, USE, AND TIME1 ARRAYS
CC*    AA       - ACTUAL USE AVAILABLE PER YEAR ARRAY
CC*    RGUS     - NUMBER OF RGUS ARRAY
CC*    TIME2    - YEAR NUMBER ASSOCIATED WITH EACH AA AND RGUS VALUE
CC*    NPTS2    - NUMBER OF ELEMENTS IN AA, RGUS, AND TIME2 ARRAYS
CC*    NAME     - TEN CHARACTER NAME OF RESOURCE
CC*
CC*    EACH OF THE DATA ARRAYS (XMA,USE,TIME1,AA,RGUS,TIME2) HAVE
CC*    AS THEIR FIRST POINT, THE VALUE FOR THE BEGINNING OF THE
CC*    FIRST BUCKET. THE REST OF THE VALUES ARE FOR THE END OF EACH
CC*    BUCKET. THEREFORE, THE NUMBER OF POINTS EQUALS THE NUMBER
CC*    OF BUCKETS PLUS ONE.
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    SCALE1
CC*    SCALE2
CC*    STEPFN
CC*    MAXMIN
CC*    LABELX
CC*    PLOT
CC*    SYMBOL
CC*    AXIS
CC*****

```

```

CC***** MAXMIN *****
CC*
CC*          SUBROUTINE MAXMIN
CC*
CC*  PURPOSE
CC*    TO FIND THE MAXIMUM AND MINIMUM VALUE IN AN ARRAY
CC*
CC*  CALLING SEQUENCE
CC*    CALL MAXMIN (XARRAY, NPTS, XMIN, XMAX)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      XARRAY - ARRAY OF VALUES
CC*      NPTS   - NUMBER OF VALUES
CC*    OUTPUT
CC*      XMIN   - MINIMUM VALUE IN XARRAY
CC*      XMAX   - MAXIMUM VALUE IN XARRAY
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** SCALE1 *****
CC*
CC*          SUBROUTINE SCALE1
CC*
CC*  PURPOSE
CC*    TO COMPUTE THE SCALE FACTORS FOR AN ARRAY OF VALUES, BASED
CC*    ON THE PLOT DIMENSIONS. THIS IS AN ISOLATION ROUTINE TO
CC*    PROVIDE COMPATIBILITY BETWEEN THE CALSPAN SCALE SUBROUTINE
CC*    AND THE STANDARD CALCOMP SCALE SUBROUTINE.
CC*
CC*  CALLING SEQUENCE
CC*    CALL SCALE1 (XARRAY, NPTS, SIZE, XMIN, DX)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      XARRAY - ARRAY OF VALUES FOR WHICH A SCALE IS TO BE COMPUTED
CC*      NPTS   - NUMBER OF ELEMENTS IN XARRAY
CC*      SIZE   - LENGTH IN FLOATING POINT INCHES AVAILABLE FOR
CC*               PLOTTING THE ARRAY
CC*    OUTPUT
CC*      XMIN   - VALUE OF FIRST ANNOTATION ON THE AXIS
CC*      DX     - SCALE FACTOR (NUMBER OF UNITS PER INCH OF PLOT)
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    MAXMIN
CC*    SCALE
CC*****

```



```

CC***** SCALE2 *****
CC*
CC*          SUBROUTINE SCALE2
CC*
CC*  PURPOSE
CC*    TO CONVERT AN ARRAY OF VALUES INTO PLOTTER INCHES
CC*
CC*  CALLING SEQUENCE
CC*    CALL  SCALE2 (XARRAY, NPTS, XMIN, DX)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT-OUTPUT
CC*    XARRAY - ARRAY OF VALUES TO BE CONVERTED
CC*    INPUT
CC*      NPTS   - NUMBER OF ELEMENTS IN XARRAY
CC*      XMIN   - MINIMUM VALUE ON PLOT AXIS
CC*      DX     - PLOT AXIS INCREMENT (UNITS PER INCH)
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** STEPFN *****
CC*
CC*          SUBROUTINE STEPFN
CC*
CC*  PURPOSE
CC*    TO PLOT A STEP FUNCTION
CC*
CC*  CALLING SEQUENCE
CC*    CALL STEPFN (XARRAY, YARRAY, NPTS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      XARRAY - X VALUES OF THE POINTS DEFINING THE STEP FUNCTION
CC*              (IN PLOTTER INCHES)
CC*      YARRAY - Y VALUES OF THE POINTS DEFINING THE STEP FUNCTION
CC*              (IN PLOTTER INCHES)
CC*      NPTS   - NUMBER OF ELEMENTS IN XARRAY AND YARRAY
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    PLOT
CC*
CC*****

```

```

CC***** LABELX *****
CC*
CC*          SUBROUTINE LABELX
CC*
CC*  PURPOSE
CC*    TO LABEL A LINE DRAWN BY SUBROUTINE STEPFN. THE X POSITION
CC*    OF THE LABEL IS SPECIFIED, AND THE Y POSITION IS COMPUTED
CC*    SO THAT IT WILL BE ON THE LINE.
CC*
CC*  CALLING SEQUENCE
CC*    CALL LABELX (X, Y, NPTS, XL, LBL, NC, CHRSZ)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      X      - X COORDINATES
CC*      Y      - Y COORDINATES
CC*      NPTS   - NUMBER OF ELEMENTS IN X AND Y
CC*      XL     - X POSITION OF LABEL
CC*      LBL    - LABEL
CC*      NC     - NUMBER OF CHARACTERS IN LABEL
CC*      CHRSZ  - CHARACTER SIZE
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*  SYMBOL
CC*
CC*****

```

#### Section 4.4

#### SUBROUTINE CROSS REFERENCE TABLE

In the table on the following pages, the column headings show the subroutine names that do the calling, and the row headings give the subroutine names that are called.

# CROSS REFERENCE USAGE CODES

## A ARGUMENT

THE SYMBOL IS A VARIABLE OR FUNCTION NAME WHICH APPEARS IN AN ARGUMENT LIST OF A CALL, SUBROUTINE, FUNCTION, OR ENTRY STATEMENT.

## D DATA INITIALIZATION

THE SYMBOL IS A VARIABLE WHICH IS INITIALIZED IN A DATA OR TYPE SPECIFICATION STATEMENT SUCH AS A COMPLEX SPECIFICATION STATEMENT.

## F EACH A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE MOST RECENTLY ASSIGNED VALUE IS ACCESSED BUT NOT CHANGED.
2. FUNCTION NAME OR ARGUMENT OF A FUNCTION WHICH APPEARS ON THE RIGHT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT OR APPEARS IN AN IF STATEMENT TEST.
3. DUMMY ARGUMENT IN A STATEMENT FUNCTION DEFINITION.

## S STORE A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE VALUE IS REPLACED BY ANOTHER VALUE.
2. FUNCTION NAME WHICH APPEARS ON THE LEFT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT.
3. NAME OF A STATEMENT FUNCTION IN THE DEFINITION OF THAT FUNCTION.

## C COMMON

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A COMMON STATEMENT OR IS THE NAME OF A LABELED COMMON BLOCK.

## E EQUIVALENCE

THE SYMBOL IS A VARIABLE WHICH APPEARS IN AN EQUIVALENCE STATEMENT.

## T TYPE SPECIFICATION

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A:

1. TYPE SPECIFICATION STATEMENT AND IS NOT INITIALIZED IN THAT STATEMENT.
2. DIMENSION OR EXTERNAL STATEMENT.

## N ENTRY POINT

THE SYMBOL IS AN ENTRY POINT DEFINED BY AN ENTRY STATEMENT IN A SUBROUTINE OR FUNCTION.

## X EXTERNAL REFERENCE

THE SYMBOL IS A SUBROUTINE OR ENTRY NAME WHICH APPEARS IN A CALL STATEMENT.

CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	MAIN	INIT	CLEAR	*BLOCK	LOOK2	PRINT1	PRINT2	PRINT3	REPT1	REPT2	PRYEAR	PRINT4
IAA1	I												
IAA2	I												
ICUS	I	C	FSC				F C			C			
IEND	I	F C	FSC				F C			C			
IPERD	I	C	FSC				F C			C			
IRDTE	I	C	C		D C			C		C		C	
ISTART	I	F C	FSC				F C			C			
KFINAL	I	C	A C					C		C		FSC	
KPR	I	F C	A FSC					F C		C		A FSC	
KPRP	I	FSC	A C					C		A F C		C	
KPRY	I	FSC	A C					C		C		F C	
KSR	I	C	SC					C		C		C	
KSRC	I	C	A SC					F C		C		C	
KSRCL	I	C	A FSC					F C		C		F C	
KSRCY	I	C	A C					C		C		FSC	
NPR	I	C	F C		D C			C		C		C	
NRDTE	I	C	C		D C			C		C		C	
XSR	I	C	F C		D C			C		C		C	
MSRC	I	C	A F C		D C			C		C		C	
MSRCL	I	C	F C		D C			C		C		C	
MYEAR	I	C	A C		D C			C		C		C	
NEXTP	I	FSC	SC							FSC			
NEXTY	I	F C	SC							C			
NPR	I	C	A FSC		D C			F C		A F C		F C	
NRGU1	I												
NRGU2	I												
NSR	I	C	FSC		D C			C		C		C	



SYMBOL	TYPE	USAGE SUMMARY											
		SRYEAR	PRINTS	RDTE	IIICOST	SUMRY	PRINT6	MRGU	WPLOTB	FPLOT	LOOKUP	PLOTU	MAXMIN
IAA1	I							SC		F C			
IAA2	I							SC		F C			
ICUS	I	C			C		C			F C	F C		
IEND	I	C			C		C			C	C		
IPERO	I	C			C		C			C	C		
IRDTE	I	C	C	FSC		A F C				C	F C		
ISTART	I	C			C					C			
KFINAL	I	FSC	C	FSC		A C				C			
KPR	I	C	C	C		C			A F C	C			
KPRP	I	C	C	C		C				C			
KPRY	I	C	C	C		C				C			
KSR	I	A FSC	F C	C		C			A F C	C			
KSRC	I	C	C	C		C				C			
KSRL	I	C	C	C		C				C			
KSRCY	I	F C	C	C		C				C			
KPR	I	C	C	C		C				C			
KRDTE	I	C	C	F C		F C				C			
KSR	I	C	C	C		C				C			
MSRC	I	C	C	C		C				C			
MSRL	I	C	C	C		C				C			
MYEAR	I	C	C	C		C				C		C	
NEXTP	I	C			C					C		C	
NEXTY	I	F C			C								
NPR	I	C	C	C		C			F C				
NRGUL	I							SC					
NRGUZ	I							SC					
MSR	I	F C	C	C		C							

CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

SCALE1 SCALE2 STEPEN LABELX

IAA1	I
IAA2	I
ICUS	I
IEND	I
IPERD	I
IRDTTE	I
ISTART	I
KFINAL	I
KPR	I
KPRP	I
KPRY	I
KSR	I
KSRC	I
KSRL	I
KSRCY	I
MPR	I
MRDTE	I
MSR	I
MSRC	I
MSRL	I
MYEAR	I
NEXTP	I
NEXTV	I
NPR	I
NRGU1	I
NRGU2	I
NSR	I

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	MAIN	INIT	CLEAR	*BLOCK	LOOK2	PRINT1	PRINT2	PRINT3	REPT1	REPT2	PRYEAR	PRINT4
NSRC	I	C	A	FSC	D	C		C	C	C	C	C	C
NSRCL	I	C	FSC		D	C		C	C	C	C	C	C
NYEAR	I	C	C		D	C		C	C	C	C	C	C
RATE	R	FSC	FSC				F	C		C	C		

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

SYMBOL	TYPE	SRYEAR	PRINTS	ROTE	IIICOST	SUMRY	PRINT6	MRGU	WPLGTB	FPLGT	LOOKUP	PLOTU	MAXMIN
NSRC	I	C	C	C	C	C				C			
NSRCL	I	C	C	C		C				C			
NYEAR	I	C	C	C		C				C			
RATE	R	C			C		F C			C		C	

CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

	SCALE1	SCALE2	STEPFN	LABELX
NSRC	I			
NSRCL	I			
NYEAR	I			
RATE	R			

## Section 4.5

### COMMON VARIABLE DEFINITIONS

The tables on the following pages define the meaning of each variable contained in each of the common blocks used by this program.



COMMON /RESRCE/ - PART 1  
PRIMARY RESOURCE DATA

\* VARIABLE \* DESCRIPTION

\* KPR(18,J) \* PRIMARY RESOURCE DATA (18 WORDS PER PR, SECOND SUBSCRIPT  
\* IS INDEXED BY PR NUMBER)

\*WORD \* CONTENTS

- \* 1 \* POINTER TO FIRST SRC USED BY THIS PR  
\* (SUBSCRIPT IN KSRCL ARRAY)
- \* 2 \* POINTER TO LAST SRC USED BY THIS PR  
\* (SUBSCRIPT IN KSRCL ARRAY)  
\* OR ZERO IF NO SRC IS USED
- \* 3 \* NUMBER OF USE UNITS WHICH AN RGU FOR THIS PR CAN  
\* PRODUCE PER YEAR
- \* 4 \* RDTE COST (DOLLARS)
- \* 5 \* RDTE PERIOD (YEARS)
- \* 6 \* INITIAL INVESTMENT COST PER RGU (DOLLARS)
- \* 7 \* RECURRING INVESTMENT COST PER RGU (DOLLARS)
- \* 8 \* RECURRING INVESTMENT COST PER YEAR (DOLLARS)
- \* 9 \* OPERATIONS AND MAINTENANCE COST PER RGU (DOLLARS)
- \* 10 \* OPERATIONS AND MAINTENANCE COST PER UNIT OF USE (DOLLARS)
- \* 11 \* PLOT SWITCH
- \* 12 \* TRANSFERRABLE SWITCH
- \* 13 \* BUCKET SIZE (CUS)
- \* 14 \* FIRST TIME USE SWITCH
- \* 15 \* CURRENT NUMBER OF RGUS REQUIRED FOR THIS PR
- \* 16 \* PR NAME (FIRST 4 CHARACTERS)
- \* 17 \* PR NAME (SECOND 4 CHARACTERS)
- \* 18 \* PR NAME (LAST 2 CHARACTERS)

\* NPR \* NUMBER OF PRIMARY RESOURCES IN THE KPR ARRAY  
\* MPR \* MAXIMUM NUMBER OF PRIMARY RESOURCES WHICH CAN BE DEFINED  
\* (DIMENSION OF SECOND SUBSCRIPT OF KPR ARRAY)

\*\*\*\*\*  
 \*  
 \* COMMON /RESRCE/ - PART 2  
 \* SECONDARY RESOURCE COMPONENT USAGE LIST  
 \*  
 \*\*\*\*\*

\* VARIABLE \* DESCRIPTION  
 \*

\*KSRCL(5,J)\* SRC USAGE DATA FOR PRIMARY RESOURCES (5 WORDS PER SRC, EACH  
 \* PR DEFINITION GIVES THE J INDEX FOR THE FIRST AND LAST SRC  
 \* WHICH IT USES)  
 \*

\* WORD \* CONTENTS  
 \*

- \* 1 \* UNITS OF USE PER RGU FOR THE PR
- \* 2 \* UNITS OF USE PER UNIT OF PR USE
- \* 3 \* POINTER TO SRC NAME IN KSRC ARRAY  
 \* (INITIALLY CONTAINS FIRST 4 CHARACTERS OF SRC NAME)
- \* 4 \* SECOND 4 CHARACTERS OF SRC NAME
- \* 5 \* LAST TWO CHARACTERS OF SRC NAME

\*  
 \*  
 \*NSRCL \* NUMBER OF ENTRIES IN KSRCL ARRAY  
 \*MSRCL \* MAXIMUM NUMBER OF ENTRIES WHICH CAN BE STORED IN THE KSRCL ARRAY  
 \* (DIMENSION OF SECOND SUBSCRIPT)  
 \*

\*\*\*\*\*  
 \*  
 \* COMMON /RESRCE/ - PART 3  
 \* SECONDARY RESOURCE DATA  
 \*  
 \*\*\*\*\*

\* VARIABLE \* DESCRIPTION  
 \*

\* KSR(17,J) \* SECONDARY RESOURCE DATA (17 WORDS PER SR, SECOND SUBSCRIPT IS  
 \* INDEXED BY SR NUMBER)  
 \*

\* WORD \* CONTENTS  
 \*

- \* 1 \* POINTER TO FIRST COMPONENT OF THIS SR  
 \* (SUBSCRIPT IN KSRC ARRAY)
- \* 2 \* POINTER TO LAST COMPONENT OF THIS SR  
 \* (SUBSCRIPT IN KSRC ARRAY)
- \* 3 \* NUMBER OF USE UNITS WHICH AN RGU FOR THIS SR CAN  
 \* PRODUCE PER YEAR
- \* 4 \* RDTE COST (DOLLARS)
- \* 5 \* RDTE PERIOD (YEARS)
- \* 6 \* INITIAL INVESTMENT COST PER RGU (DOLLARS)
- \* 7 \* RECURRING INVESTMENT COST PER RGU (DOLLARS)
- \* 8 \* RECURRING INVESTMENT COST PER YEAR (DOLLARS)
- \* 9 \* OPERATIONS AND MAINTENANCE COST PER RGU (DOLLARS)
- \* 10 \* OPERATIONS AND MAINTENANCE COST PER UNIT OF USE (DOLLARS)
- \* 11 \* PLOT SWITCH
- \* 12 \* TRANSFERRABLE SWITCH
- \* 13 \* FIRST TIME USE SWITCH
- \* 14 \* CURRENT NUMBER OF RGUS REQUIRED FOR THIS SR
- \* 15 \* SR NAME - FIRST FOUR CHARACTERS
- \* 16 \* SR NAME - SECOND FOUR CHARACTERS
- \* 17 \* SR NAME - LAST TWO CAHRACTERS

\* NSR \* NUMBER OF SECONDARY RESOURCES IN THE KSR ARRAY  
 \* MSR \* MAXIMUM NUMBER OF SECONDARY RESOURCES WHICH CAN BE DEFINED  
 \* (DIMENSION OF SECOND SUBSCRIPT OF KSR ARRAY)  
 \*

```

*****
*
*          COMMON /RESRCE/ - PART 4
*        SECONDARY RESOURCE COMPONENTS
*
*****

```

```

*
* VARIABLE *   DESCRIPTION
*
*****
* KSR(3,J) * SECONDARY RESOURCE COMPONENT NAMES
*           * (3 WORDS PER 10 CHARACTER NAME)
* NSRC      * NUMBER OF SRC NAMES IN THE KSRC ARRAY
* MSRC      * MAXIMUM NUMBER OF SRCS WHICH CAN BE DEFINED
*           * (DIMENSION OF SECOND SUBSCRIPT OF KSRC ARRAY)
*
*****

```

\*\*\*\*\*  
 \*  
 \* COMMON /RESRCE/ - PART 5  
 \* RESOURCE USAGE TABLES  
 \*  
 \*\*\*\*\*

\* VARIABLE \* DESCRIPTION  
 \*\*\*\*\*

\* KPRP(3,J) \* PR USE DATA FOR PERIODIC REPORT  
 \* (SECOND SUBSCRIPT MUST BE DIMENSIONED AT THE MAXIMUM NUMBER  
 \* OF PRIMARY RESOURCES ALLOWED, WHICH IS GIVEN BY VARIABLE MPR)  
 \*\*\*\*\*

\* 1 PEAK USE PER BUCKET  
 \* 2 NUMBER OF BUCKETS DURING WHICH THIS RESOURCE WAS USED  
 \* 3 TOTAL USE  
 \*\*\*\*\*

\* KPRY(3,J) \* PR USE DATA FOR YEARLY REPORT  
 \* (J DIMENSION MUST BE THE SAME AS THAT OF KPRP ARRAY)  
 \*\*\*\*\*

\* 1 PEAK USE PER BUCKET  
 \* 2 NUMBER OF BUCKETS DURING WHICH THIS RESOURCE WAS USED  
 \* 3 TOTAL USE  
 \*\*\*\*\*

\* KSRCY(K) \* SRC USE DATA FOR THE YEARLY REPORT  
 \* (MUST BE DIMENSIONED AT THE MAXIMUM NUMBER OF SECONDARY  
 \* RESOURCE COMPONENTS ALLOWED, WHICH IS GIVEN BY VARIABLE MSRC)  
 \*  
 \* EACH ELEMENT CONTAINS THE TOTAL USE FOR THAT SRC DURING THE  
 \* YEAR  
 \*\*\*\*\*

\*\*\*\*\*  
 \*  
 \* COMMON /RESRCE/ - PART 6  
 \* FINAL COST CUMMARY  
 \*  
 \*\*\*\*\*

* VARIABLE	* DESCRIPTION
* KFINAL(4,J)	* TOTAL COST FOR EACH CATEGORY IN EACH YEAR
	* 1 ROTE COST
	* 2 INITIAL INVESTMENT COST
	* 3 RECURRING INVESTMENT COST
	* 4 OPERATIONS AND MAINTENANCE COST
* NYEAR	* NUMBER OF YEARS FOR WHICH COST DATA HAS BEEN STORED IN THE KFINAL ARRAY
* MYEAR	* MAXIMUM NUMBER OF YEARS ALLOWED (DIMENSION OF SECOND SUBSCRIPT OF KFINAL)
* IRDTE(K)	* ROTE COST TABLE FOR NEGATIVE YEARS
	* IRDTE(1) WOULD CONTAIN THE ROTE COST FOR YEAR ZERO
	* IRDTE(2) WOULD CONTAIN THE ROTE COST FOR YEAR -1
* MRDTE	* DIMENSION OF IRDTE ARRAY

\*\*\*\*\*



```

*****
*
*                                COMMON /PARMS/
*
*****
*  VARIABLE  *  DESCRIPTION
*
*****
*  ISTART   *  START TIME FOR PHASE 4 TO START GENERATING REPORTS
*  IEND     *  END TIME FOR PHASE 4 REPORTS
*  IPERD    *  TIME INTERVAL BETWEEN PERIODIC REPORTS
*  RATE     *  INFLATION RATE (RATIO - .1 = 10 PERCENT)
*  ICUS     *  NUMBER OF CALENDAR UNITS PER YEAR
*  NEXTP    *  TIME OF NEXT PERIODIC REPORT
*  NEXTY    *  TIME OF NEXT YEARLY REPORT
*
*****

```

```

*****
*
*          COMMON /RGU/
*      COMMON TO HOLD RGU PLOT DATA
*
*****

```

```

* VARIABLE * DESCRIPTION
*
*****

```

```

* NRGU1(I,J) * NUMBER OF RGUS ON HAND FOR EACH PRIMARY RESOURCE DURING
*             * EACH YEAR
* IAA1(I,J)  * ACTUAL USE AVAILABLE FOR EACH PRIMARY RESOURCE DURING
*             * EACH YEAR
* NRGU2(I,J) * NUMBER OF RGUS ON HAND FOR EACH SECONDARY RESOURCE DURING
*             * EACH YEAR
* IAA2(I,J)  * ACTUAL USE AVAILABLE FOR EACH SECONDARY RESOURCE DURING
*             * EACH YEAR

```

```

*             * FOR ALL OF THE ABOVE ARRAYS

```

```

*             * I SUBSCRIPT IS INDEXED ON YEAR NUMBER

```

```

*             * J SUBSCRIPT IS INDEXED ON RESOURCE NUMBER
*
*****

```

## Section 4.6

### COMMON VARIABLE CROSS REFERENCE TABLE

The table on the following pages shows how each subroutine uses each common variable. The subroutine names are printed across the top of the table, and the variable names down the left side.

SUBROUTINE CROSS REFERENCE SUMMARY \*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

	MAIN	INIT	CLEAR	*BLOCK	LOOK2	PRINT1	PRINT2	PRINT3	REPT1	REPT2	PRYEAR	PRINT4
CLEAR		X							X			
FPLOT	X											
IICOST											X	
INIT	X											
LABELX												
LOOKUP		X										
LOOK2		X										
MAXMIN												
PLOTU												
PRINT1	X											
PRINT2	X											
PRINT3	X											
PRINT4											X	
PRINT5												
PRINT6												
PRYEAR									X			
RDTE											X	
REPT1	X											
REPT2	X											
SCALE1												
SCALE2												
SRYEAR										X		
STEPFN												
SUMRY	X											
WPLOTB	X											
WRGU											X	

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

ROUTINE OR ENTRY	SRYEAR	PRINT5	ROTE	IICOST	SUMRY	PRINT6	WRGU	WPLOT6	FPLOT	LOOKUP	PLOTU	MAXMIN
CLEAR												
FPLOT												
IICOST	X											
INIT												
LABELX											X	
LOOKUP												
LOOK2												
MAXMIN											X	
PLOTU									X			
PRINT1												
PRINT2												
PRINT3												
PRINT4												
PRINT5	X											
PRINT6					X							
PRYEAR												
ROTE	X											
REPT1												
REPT2												
SCALE1											X	
SCALE2											X	
SYEAR												
STEPFN											X	
SUMRY												
WPLOTB	X											
WRGU	X											

SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASE 4 \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

SCALE1 SCALE2 STEPFN LABELX

CLEAR

FPLOT

ITCOST

INIT

LABELX

LOOKUP

LOOK2

MAXMIN

PLOTU

PRINT1

PRINT2

PRINT3

PRINT4

PRINT5

PRINT6

PRYEAR

ROTE

REPT1

REPT2

SCALE1

SCALE2

SRYEAR

STEPFN

SUNRY

WPLOTB

WRGU

x



Section 4.7  
TEMPORARY FILES

Two temporary files are created for the purposes of sorting data for plotting. The following figures describe the records on these files.

```

*****
*
*               FILE 51 DESCRIPTION
*          PRIMARY RESOURCE PLOT BUCKET FILE
*
*****

```

```

*
*  THIS UNFORMATTED (BINARY) FILE CONTAINS RECORDS OF THE FOLLOWING FORMAT
*
*****

```

```

*      *
* WORD *  CONTENTS
*      *
*****

```

- ```

*      1 * PRIMARY RESOURCE NUMBER
*      2 * BUCKET END TIME (CUS)
*      3 * NUMBER OF UNITS OF THE RESOURCE USED DURING THIS BUCKET
*      4 * MAXIMUM NUMBER OF UNITS OF THE RESOURCE AVAILABLE DURING THIS BUCKET
*      *
*****

```

\*\*\*\*\*  
 \*  
 \* FILE 52 DESCRIPTION \*  
 \* SECONDARY RESOURCE PLOT BUCKET FILE \*  
 \*  
 \*\*\*\*\*

\* THIS UNFORMATTED (BINARY) FILE CONTAINS RECORDS OF THE FOLLOWING FORMAT \*  
 \*\*\*\*\*

\* WORD \* CONTENTS \*  
 \*\*\*\*\*

\* 1 \* SECONDARY RESOURCE NUMBER \*  
 \* 2 \* END TIME OF THIS YEAR IN CALENDAR UNITS \*  
 \* 3 \* NUMBER OF UNITS OF THE RESOURCE USED DURING THIS YEAR \*  
 \* 4 \* MAXIMUM NUMBER OF UNITS OF THE RESOURCE AVAILABLE DURING THIS YEAR \*  
 \* \* (NOTE - THIS IS THE SAME AS ACTUAL AVAILABLE FOR SR) \*  
 \*\*\*\*\*

Section 5.0  
PHASE 5 PROGRAMMER'S GUIDE

INTRODUCTION

The purpose of Phase 5 of TRAM is to report on the usage of trainees and on the time lags that occur in the training system.

This manual is intended to aid the programmer in the operation and modification of the computer program. It is assumed that the reader of this manual is already familiar with the contents of Technical Memorandum SAT-5, TRAM User's Guide.

## Section 5.2

### PROGRAM DESCRIPTION

Phase 5 reads the time ordered records from the source/lag file and stores a summary of these events in two separate areas, one for the periodic reports, and the other for the yearly reports. These reports are produced at the specified intervals from this stored information. When the end time that was specified on the parameter card is reached, the program outputs a set of final reports to cover the period from the last yearly report to the end of the run.

### Section 5.3

#### SUBPROGRAM DESCRIPTIONS

This section contains the descriptions of the individual subprogram's that comprise phase 5 of the TRAM program. The description for each subprogram consists of a statement of the purpose of the routine, the calling sequence, a description of its parameters, the method used, and a list of the subprograms required. A high level flowchart, which shows the logical decision points and the processing accomplished, is also included for each of the major subprograms.

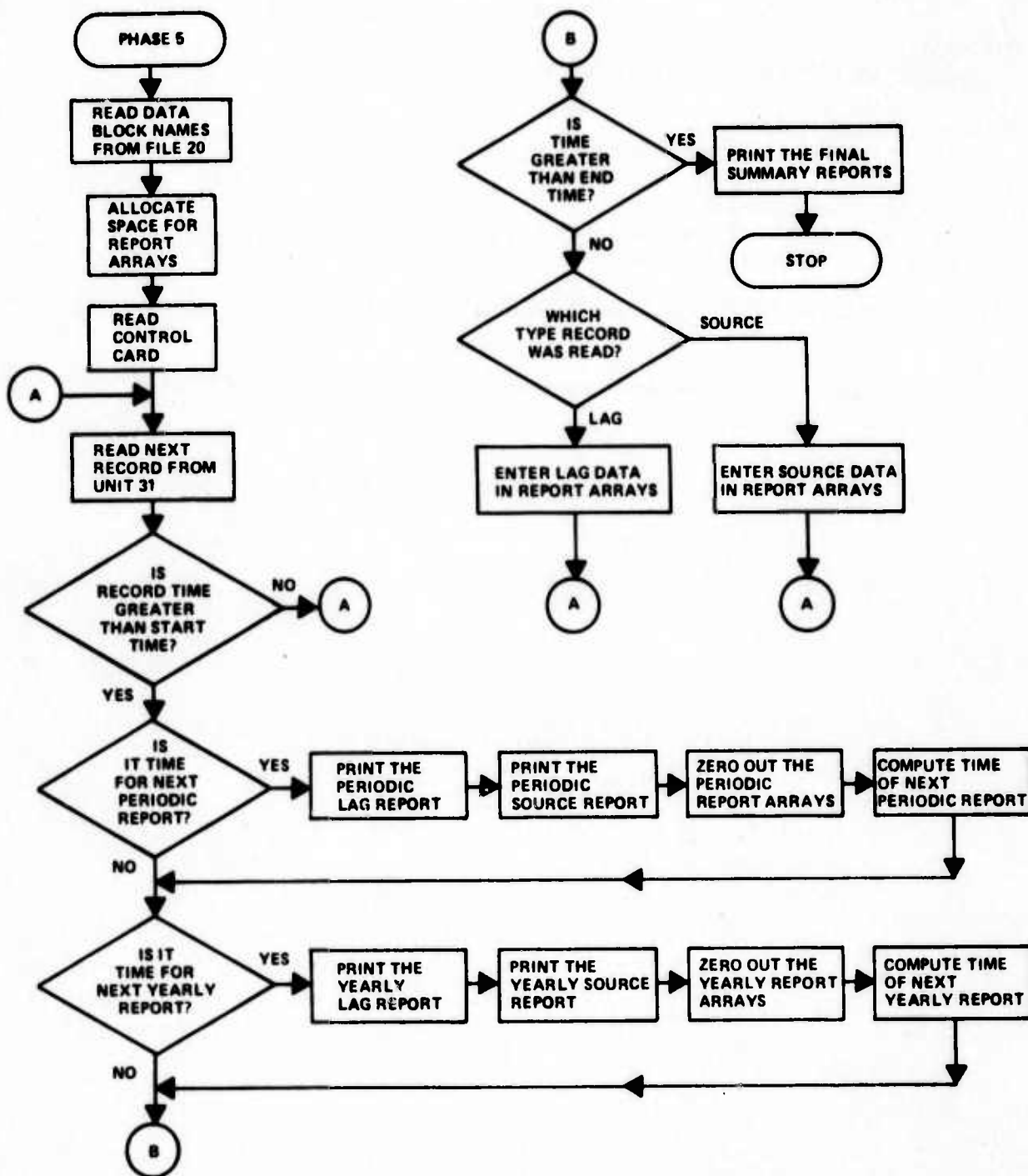
Two subroutines (RDNAME and NAME) are not shown here, but are documented in the phase 3 programmers' guide.



```

CC***** PHASE5 *****
CC*
CC*          SUBROUTINE PHASE5
CC*
CC*  PURPOSE
CC*    TO OUTPUT THE LAG REPORTS AND THE SOURCE REPORTS FROM THE
CC*    RESULTS OF PHASE3.
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    RDNAM
CC*    CLEAR
CC*    REPRT1
CC*    REPRT2
CC*
CC*****

```



PHASE 5 MAIN PROGRAM

```

CC***** CLEAR *****
CC*
CC*
CC*          SUBROUTINE CLEAR
CC*
CC*  PURPOSE
CC*    TO CLEAR AN ARRAY TO ZERO
CC*
CC*  CALLING SEQUENCE
CC*    CALL  CLEAR (IARRAY, NWDS)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    IARRAY - ARRAY TO BE CLEARED
CC*    NWDS   - NUMBER OF ELEMENTS IN IARRAY TO BE CLEARED
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NONE
CC*****

```

```

CC***** REPRT1 *****
CC*
CC*          SUBROUTINE REPRT1
CC*
CC*  PURPOSE
CC*    TO PRINT THE LAG REPORT
CC*
CC*  CALLING SEQUENCE
CC*    CALL  REPRT1 (ICODE, ISTRT, IEND)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICODE - INDICATES REPORT TYPE
CC*              1 PERIODIC
CC*              2 YEARLY
CC*              3 FINAL SUMMARY
CC*      ISTRT - START TIME OF THE REPORT
CC*      IEND  - END TIME OF THE REPORT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NAME
CC*
CC*****

```

```

CC***** REPRT2 *****
CC*
CC*          SUBROUTINE REPRT2
CC*
CC*  PURPOSE
CC*    TO PRINT THE SOURCE REPORT
CC*
CC*  CALLING SEQUENCE
CC*    CALL  REPRT2 (ICODE,ISTRT,IEND)
CC*
CC*  DESCRIPTION OF PARAMETERS
CC*    INPUT
CC*      ICODE - INDICATES REPORT TYPE
CC*              1 PERIODIC
CC*              2 YEARLY
CC*              3 FINAL SUMMARY
CC*      ISTRT - START TIME OF THE REPORT
CC*      IEND  - END TIME OF THE REPORT
CC*
CC*  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*    NAME
CC*
CC*****

```

## Section 5.4

### SUBROUTINE CROSS REFERENCE TABLE

In the table on the following page, the column headings show the names of the subroutines that do the calling, and the row headings give the names of the subroutines that are called.



SUBROUTINE CROSS REFERENCE SUMMARY C\*\*\*\*\* PHASES \*\*\*\*\*

USAGE SUMMARY

ROUTINE  
OR ENTRY

|       | MAIN | #BLOCK | CLEAR | REPT1 | REPT2 |
|-------|------|--------|-------|-------|-------|
| CLEAR | X    |        |       |       |       |
| REPT1 | X    |        |       |       |       |
| REPT2 | X    |        |       |       |       |

Section 5.5  
COMMON VARIABLE DEFINITIONS

The tables on the following pages define the meaning of each variable contained in each of the common blocks used by this program.



```

*****
*
*   COMMON /SOURCE/ - DATA FOR SOURCE REPORTS
*
*****
*
*   VARIABLE *   DESCRIPTION
*
*****
*   IPS1 *   BASE POINTER TO SOURCE DATA FOR COURSES
*   NS1 *   NUMBER OF COURSES
*   IPS2 *   BASE POINTER TO SOURCE DATA FOR SOURCES
*   NS2 *   NUMBER OF SOURCES
*
*   *ISRCE(I,8) * SOURCE DATA - 8 WORDS PER ENTRY
*
*   *****
*   *
*   *   WORD *   CONTENTS
*   *
*   *****
*   *
*   *   1 *   NUMBER OF TRAINEES USED DURING THIS PERIOD
*   *   2 *   TOTAL TRAINING TIME FOR TRAINEES USED DURING THIS PERIOD
*   *   3 *   TOTAL NUMBER OF TRAINEES USED TO DATE
*   *   *   (FOR PERIODIC REPORT)
*   *   4 *   TOTAL TRAINING TIME TO DATE (FOR PERIODIC REPORT)
*   *   5 *   NUMBER OF TRAINEES USED DURING THIS YEAR
*   *   6 *   TOTAL TRAINING TIME FOR TRAINEES USED DURING THIS YEAR
*   *   7 *   TOTAL NUMBER OF TRAINEES USED TO DATE
*   *   *   (FOR YEARLY REPORT)
*   *   8 *   TOTAL TRAINING TIME TO DATE (FOR YEARLY REPORT)
*   *
*   *****
*
*   *MSRCE *   MAXIMUM NUMBER OF ENTRIES WHICH CAN BE STORED IN ISRCE ARRAY
*   *   (DIMENSION OF FIRST SUBSCRIPT)
*
*****

```

## Section 5.6

### COMMON VARIABLE CROSS REFERENCE TABLE

The table on the following pages shows how each subroutine uses each common variable. The subroutine names are printed across the top of the table, and the variable names down the left side.

# CROSS REFERENCE USAGE CODES

## A ARGUMENT

THE SYMBOL IS A VARIABLE OR FUNCTION NAME WHICH APPEARS IN AN ARGUMENT LIST OF A CALL, SUBROUTINE, FUNCTION, OR ENTRY STATEMENT.

## D DATA INITIALIZATION

THE SYMBOL IS A VARIABLE WHICH IS INITIALIZED IN A DATA OR TYPE SPECIFICATION STATEMENT SUCH AS A COMPLEX SPECIFICATION STATEMENT.

## F FEICH A VALUE

THE SYMBOL IS A:  
1. VARIABLE WHOSE MOST RECENTLY ASSIGNED VALUE IS ACCESSED

2. BUT NOT CHANGED.
3. FUNCTION NAME OR ARGUMENT OF A FUNCTION WHICH APPEARS ON THE RIGHT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT OR APPEARS IN AN IF STATEMENT TEST.
4. DUMMY ARGUMENT IN A STATEMENT FUNCTION DEFINITION.

## S STORE A VALUE

THE SYMBOL IS A:

1. VARIABLE WHOSE VALUE IS REPLACED BY ANOTHER VALUE.
2. FUNCTION NAME WHICH APPEARS ON THE LEFT SIDE OF AN EQUAL SIGN IN AN ASSIGNMENT STATEMENT.
3. NAME OF A STATEMENT FUNCTION IN THE DEFINITION OF THAT FUNCTION.

## C COMMON

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A COMMON STATEMENT OR IS THE NAME OF A LABELED COMMON BLOCK.

## E EQUIVALENCE

THE SYMBOL IS A VARIABLE WHICH APPEARS IN AN EQUIVALENCE STATEMENT.

## T TYPE SPECIFICATION

THE SYMBOL IS A VARIABLE WHICH APPEARS IN A:

1. TYPE SPECIFICATION STATEMENT AND IS NOT INITIALIZED IN THAT STATEMENT.
2. DIMENSION OR EXTERNAL STATEMENT.

## N ENTRY POINT

THE SYMBOL IS AN ENTRY POINT DEFINED BY AN ENTRY STATEMENT IN A SUBROUTINE OR FUNCTION.

## X EXTERNAL REFERENCE

THE SYMBOL IS A SUBROUTINE OR ENTRY NAME WHICH APPEARS IN A CALL STATEMENT.



CROSS REFERENCE SUMMARY \*\*\*\*\* PHASES \*\*\*\*\*

USAGE SUMMARY

SYMBOL TYPE

|       |   | MAIN  | *BLOCK | CLEAR | REPT1 | REPT2 |
|-------|---|-------|--------|-------|-------|-------|
| ILAG  | I | A FSC | D C    |       | F C   |       |
| IPL1  | I | FSC   | C      |       | F C   |       |
| IPL2  | I | FSC   | C      |       | F C   |       |
| IPL3  | I | FSC   | C      |       | F C   |       |
| IPS1  | I | FSC   | C      |       |       | F C   |
| IPS2  | I | FSC   | C      |       |       | F C   |
| ISRCE | I | A FSC | D C    |       |       | F C   |
| MLAG  | I | A F C | D C    |       | C     |       |
| MSRCE | I | A F C | D C    |       |       | C     |
| NL1   | I | FSC   | C      |       | F C   |       |
| NL2   | I | FSC   | C      |       | F C   |       |
| NL3   | I | FSC   | C      |       | F C   |       |
| NS1   | I | FSC   | C      |       |       | F C   |
| NS2   | I | FSC   | C      |       |       | F C   |

Section 6.0  
MERGE PROGRAM PROGRAMMER'S GUIDE  
INTRODUCTION

The purpose of this program is to merge the original resources file from Phase 2 and the unused resources file from Phase 3 into a single resource use file for input to Phase 4.

This manual is intended to aid the programmer in the operation and modification of the computer program.

## Section 6.2

### PROGRAM DESCRIPTION

The merge program reads matched pairs of records from units 23 and 33, and outputs single records containing the data from the original pair. The records on unit 23 contain the quantity of the resource available during the bucket, and the records on unit 33 contain the quantity remaining at the end of the bucket. Both of these files are sorted by time and resource number, which are used as keys to the merge operation.

The program keeps track of the time at which the next record should be encountered for each resource, based on its bucket size (obtained from unit 24.) This enables any missing records on unit 23 to be detected. A missing record on this file indicates that the resource is no longer available. When this occurs, the program writes a record to the output file with both the quantity available and the quantity remaining set to zero. If the resource becomes available again later, the merge program will continue processing it. A missing record on unit 33 indicates that all of the resource has been used, so the quantity remaining is set to zero and a record is written to the output file.

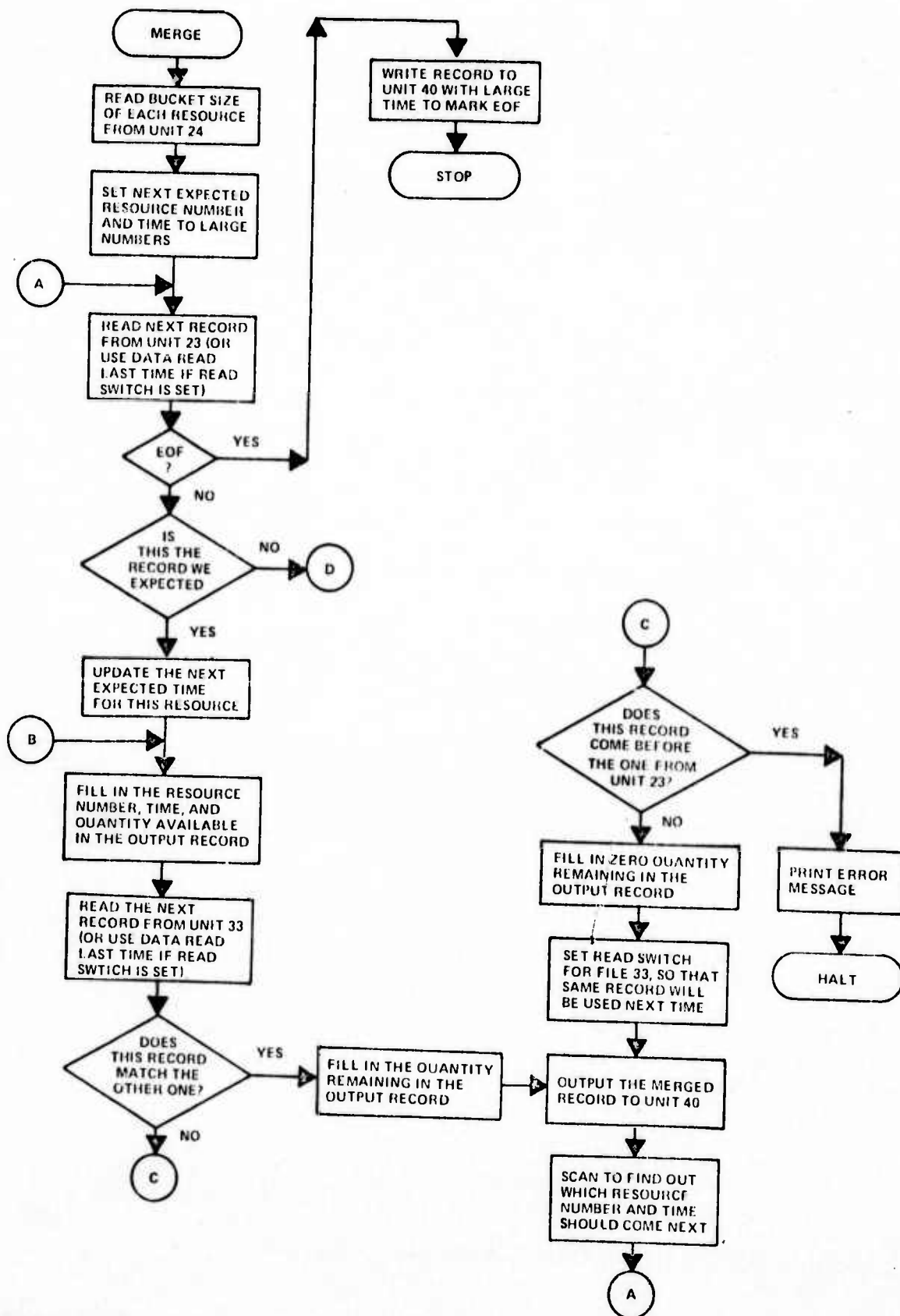
This merge operation continues until an end of file is encountered on unit 23. An extra record is then written to the output file to signal the end of file to phase 4. This record has all fields set to zero except for the bucket time, which is filled in with a large number.

The merge program consists of a main program, which does not require any subroutines or common blocks. The documentary prologue from this program is shown on the next page, and a high level flowchart is included in the pages following that.

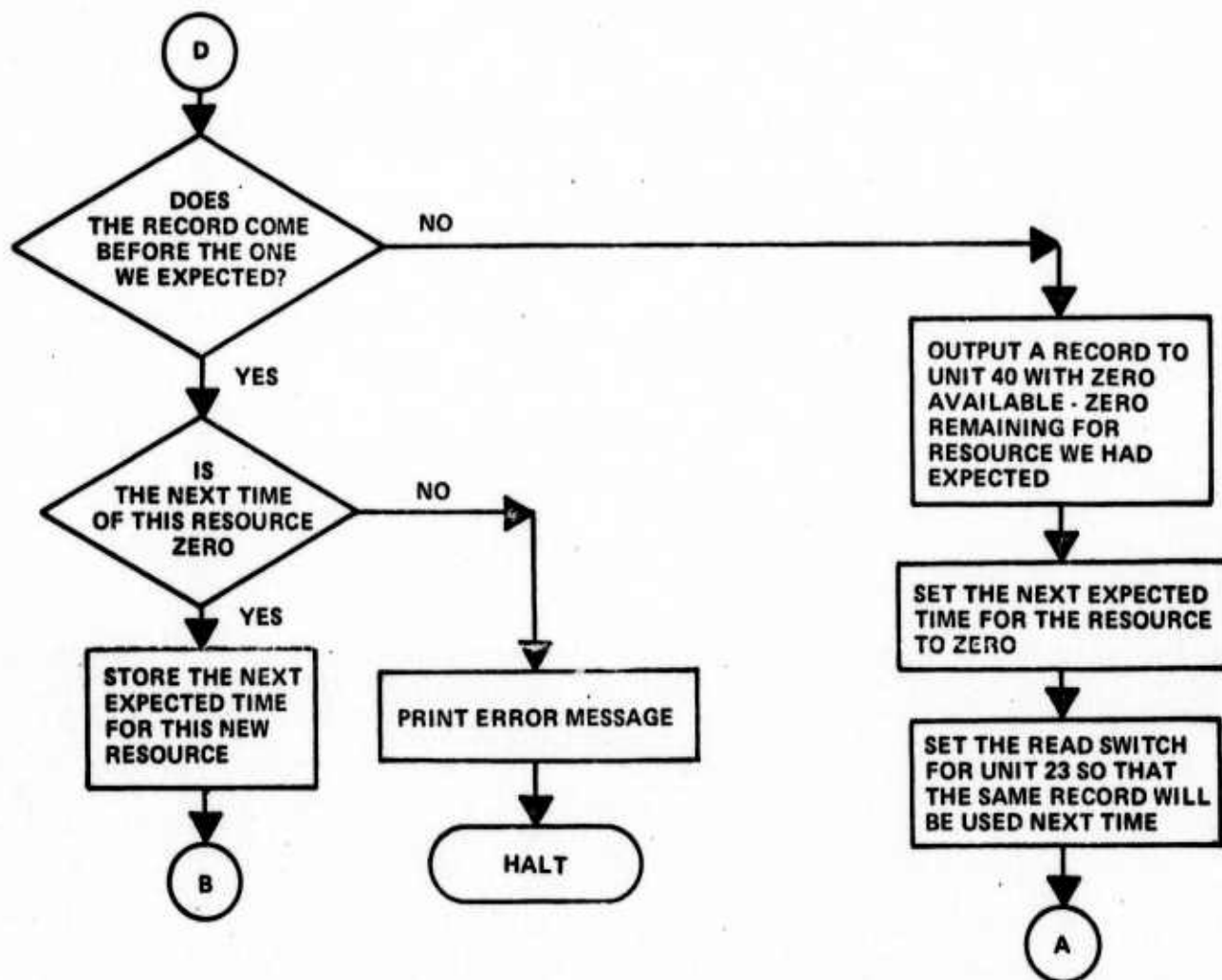
```

CC***** MERGE *****
CC*
CC*                                MERGE PROGRAM
CC*
CC* PURPOSE
CC*   TO READ THE RESOURCE QUANTITY AVAILABLE FILE FROM TRAM STEP 2
CC*   AND THE RESOURCE QUANTITY REMAINING FILE FROM TRAM STEP 3,
CC*   AND TO MERGE THEM INTO A SINGLE RESOURCE USE FILE FOR INPUT
CC*   TO TRAM STEP 4.
CC*
CC* INPUT FILES
CC*   1 TRANSFERRED PRIMARY RESOURCES FILE (TPR) - FROM TRAM STEP 2
CC*     ON FORTRAN LOGICAL UNIT 24
CC*     RECORD 1 - NUMBER OF RESOURCES
CC*             2 - RESOURCE NAMES
CC*             3 - NUMBER OF CALENDAR UNITS PER YEAR
CC*             4 - RESOURCE BUCKET SIZES
CC*
CC*   2 RESOURCE QUANTITY AVAILABLE FILE - FROM TRAM STEP 2
CC*     ON FORTRAN LOGICAL UNIT 23
CC*     EACH RECORD CONTAINS THE FOLLOWING
CC*       1 TIME
CC*       2 RESOURCE NUMBER
CC*       3 QUANTITY AVAILABLE
CC*       - THESE RECORDS HAVE BEEN SORTED ON TIME AND RESOURCE NUMBER
CC*       - RECORDS WILL NOT BE PRESENT FOR TIMES WHEN THE RESOURCE
CC*         IS NOT AVAILABLE
CC*
CC*   3 RESOURCE QUANTITY REMAINING FILE - FROM TRAM STEP 3
CC*     ON FORTRAN LOGICAL UNIT 33
CC*     EACH RECORD CONTAINS THE FOLLOWING
CC*       1 TIME
CC*       2 RESOURCE NUMBER
CC*       3 QUANTITY REMAINING
CC*       - THESE RECORDS HAVE BEEN SORTED ON TIME AND RESOURCE NUMBER
CC*       - RECORDS WILL NOT BE PRESENT FOR TIMES WHEN ALL PRESENT
CC*         UNITS OF THE RESOURCE HAVE BEEN USED
CC*
CC* OUTPUT FILE
CC*   RESOURCE USE FILE
CC*   ON FORTRAN LOGICAL UNIT 40
CC*   EACH RECORD CONTAINS THE FOLLOWING
CC*     1 TIME
CC*     2 RESOURCE NUMBER
CC*     3 MAXIMUM QUANTITY AVAILABLE IN THIS BUCKET
CC*     4 QUANTITY REMAINING AT THE END OF THIS BUCKET
CC*
CC* SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
CC*   NONE
CC*
CC*****

```



MERGE PROGRAM



MERGE PROGRAM - CONTINUED



### Section 6.3

#### DESCRIPTION OF INPUTS

The only inputs to the merge program are these binary files passed from the previous TRAM job steps:

- 23      (original resources file from phase 2)
- 24      (resource information from phase 2)
- 33      (unused resources file from phase 3)

A description of each of these files is contained in the programmers guide for the phase which creates it.

## Section 6.4

### DESCRIPTION OF OUTPUTS

The main output of this program is the binary file written to unit 40 for passage to phase 4. The contents of this file are described in the program's prologue, which is shown in section 2. The only other outputs are the printed error messages, or the "MERGE COMPLETED" message, which is printed to indicate that no errors were detected.

Section 7.0  
TROLIE PROGRAMMER'S GUIDE

7.1     Introduction

This guide is intended to supplement the user's guide for TROLIE in Technical Memorandum SAT-5, TRAM User's Guide. TROLIE consists of five parts. The INPUT subroutine reads inputs, documents the inputs and writes the resource name records in Unit 2. The MAIN program computes the resource use and writes the resource use records on Unit 1. Subroutines TAB and WTAB are table writers. TAB has an extra argument for writing column headings. BLOCK DATA clears arrays and introduces some literal data.

7.2     Subroutines

The section which follows contains descriptions of the subprograms, commons and output data sets.

Subroutine INPUT

The inputs are read in accordance with Table 1. As each major set of cards are read, output is created to document their values. Individual deliveries are not documented. A program error stop will occur if array sizes are exceeded (label 100), if a delivery is attempted to an undefined air base (label 100), or if there are an insufficient number of input cards (label 101). Input also writes the resource names record on unit 2 for use in Phase 4 of TRAM.

Subroutine BLOCK DATA

BLOCK DATA contains names for printout purposes and clears arrays used in MAIN. The arrays cleared by BLOCK DATA must be cleared because initial zeros are assumed.

Subroutine WTAB(NDATA, NCOLS, NLINES, COLLAB, LINLAB, TOPLAB, ISHIFT, KD, NAMES)  
and

Subroutine TAB(NDATA, NCOLS, NLINES, COLLAB, LINLAB, TOPLAB, ISHIFT, KD)

Arguments are:

|        |                                           |
|--------|-------------------------------------------|
| NDATA  | a two-dimensional array to be printed out |
| NCOLS  | number of columns                         |
| NLINES | number of lines                           |

COLLAB column label  
 LINLAB row label  
 TOPLAB overall label  
 ISHIFT index offset for rows  
 KD first dimension of NDATA  
 NAMES list of column headings

WTAB and TAB write tables 50 lines long by 10 columns wide with a row index.  
 WTAB has column headings, TAB has only indices.

### MAIN Program

Most of the computation is performed in the MAIN program. MAIN consists of a number of sections identifying the loops over time using the index IY (for year index).

- Loop on 100 This loop computes the training demands
- Loop on 300 This loop writes the results of the 100 loop
- Loop on 200 This loop computes the PMT demands
- Loop on 340 This loop writes the results of the 200 loop
- Loop on 400 This loop selects the source for the CCTS demands

The resource and track use reports are then written.

- Loop on 600 This loop computes the resource use and writes the resource use records in the proper order for Phase 4 of TRAM on Unit 1.

The final step is a listing of the resources used and the generation of the end of file record.

### 7.3 NOTES

In the loop on Table 100, NNCPY (IY + 1) contains the number of crews currently in the system. On the next pass through the loop, this is used to compute the number of new crews required that year.

The resource use date (IDTCU) is 2 calendar units less than the end of the year. Normally the number of calendar units per year is 1500 so this should not present a problem.

#### 7.4 COMMONS

The contents of the commons are indicated in Tables 7.1, 7.2 and 7.3.

Table 7.1

Common REALS -

Real Data Initialized by INPUT and BLOCK DATA

| <u>Variable</u> | <u>Definition</u>                | <u>Defined</u> |
|-----------------|----------------------------------|----------------|
| AR              | Attention Ratio                  | Input          |
| CR              | Crew Ratio                       | Input          |
| PUPR            | Copilot Upgrade Ratio            | Input          |
| ANAME (5)       | 'PLTS'                           | Block Data     |
|                 | 'CPLT'                           | Block Data     |
|                 | 'OSO '                           | Block Data     |
|                 | 'DSO '                           | Block Data     |
|                 | 'XTRA'                           | Block Data     |
| RESNAM(3,80)    | Resource Names<br>(3 words each) | Input          |

Table 7.2

Common ICS -

Integer Constants (All defined in INPUT)

|        |                               |
|--------|-------------------------------|
| NY     | Number of Years               |
| NYO    | Date of Year 1                |
| NB     | Number of Bases               |
| NS     | Number of Sources             |
| NT     | Number of Tracks              |
| NR     | Number of Resources           |
| IDELAY | Attrition Delay               |
| NCU    | Number of Calendar Units/Year |
| ITAPE  | Data Set Flag                 |

Table 7.3  
Common INTAR -  
Integer Arrays

Except for IDT and ISTAB, all names are of the form XXXPPY where XXX is arbitrary, P per "per" and YY is one or two suffix letters which indicate the dependencies as follows:

B Air Bases  
P Position  
S Source  
R Resource  
T Tracks  
Y Years

| <u>Variable</u> | <u>Definition</u>                                              | <u>Defined</u> |
|-----------------|----------------------------------------------------------------|----------------|
| NACBY           | No. of aircraft per base year year                             | Main           |
| NADBY           | No. of deliveries per base each year                           | Input          |
| NACPY           | No. of aircraft in system each year                            | Main           |
| NADPY           | No. of deliveries each year                                    | Input          |
| NCPBY           | No. of crews at each base each year                            | Main           |
| NCPY            | No. of crews in system each year                               | Main           |
| NNCPY           | No. of new crews trained each year                             | Main           |
| NRCPY           | No. of replacement crews trained each year                     | Main           |
| NTTDPY          | No. of total trainee demands each year                         | Main           |
| NXCPY           | No. of extra pairs each year                                   | Main           |
| ILTID           | Lower PMT track list index for each base                       | Input          |
| IUTID           | Upper PMT track list index for each base                       | Input          |
| IDT             | Track index list                                               | Input          |
| NTPTY           | No. of trainees in each track each year                        | Main           |
| NPPY            | No. of preps trained each year                                 | Main           |
| ISTAB           | Source for each CCTS track                                     | Input          |
| NSPSY           | No. of trainees available from each source each year           | Input          |
| IUSPY           | No. of trainees drawn from each source each year               | Main           |
| INSPTY          | No. of unit trainees in each track each year                   | Main           |
| IRUPRY          | Amount of each resources used each year                        | Main           |
| IRCPRT          | Amount of each resources used by a unit trainee in each track. | Input          |



## 7.5 REPORTS

Sample outputs are contained in the programmer's guide. The reports are:

1. Parameters - The first 2 cards
2. Air Base List - Base names, indices, and PMT track lists
3. Delivery List - Years by bases
4. CCTS Track List - Sources index for each CCTS track
5. Source Availability - Trainees available years by sources
6. Resource Use Data - Resources used by unit trainees - tracks by resources
7. CCTS Summary - Yearly A/C deliveries, total A/C deployment, new crews, replacement crews, total crews in the system, pilot upgrades, total full crew training, and extras pairs training
8. Detailed Base Delivery List
9. PMT Track Trainees - Number of PMT unit trainees per track, years by track
10. Source Use Matrix - Use of trainees from each source, years by source
11. Track Use - Trainees taught by years and track
12. Resource Use - Resources used, years by resource.

## 7.6 DATA SET OUTPUT

Two files are produced.

FORTRAN Unit 1 contains resource use records. The records must be sorted by date and resources number. Each logical record contains:

- Date in CUs
- Resource number
- Resource originally available
- Resource remaining

The resource originally available is nominally 99999 units. A final record with the time 999999 is produced as an end of file record.

FORTRAN Unit 2 is the resource name file. It contains:

Record

- 1 Number of resources
- 2 Resource names
- 3 Number of calendar units per year
- 4 Bucket size for each resource (=NCU for TROLIE)